VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2010 Drahomír Kalman

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Prezentace vlastností firewallu
Presentation of firewall properties

2010 Drahomír Kalman

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 5. 5. 2010 _____

Drahomír Kalman

Poděkování

Děkuji Ing. Pavlu Nevludovi za hodnotné rady a odborné vedení během mé práce.

Abstrakt

Tato diplomová práce se věnuje zabezpečení sítí pomocí firewallu. Jsou zde popsány jednotlivé technologie firewallů a jejich historický vývoj. Dále jsou navržena dvě základní zapojení firewallu k zabezpečení sítě a popis jejich konfigurace. Vlastnosti těchto zapojení jsou ověřeny pomocí penetračního testování.

Abstract

This thesis is dedicated to network security via firewall. There are described various technologies of firewalls and their historical development. There are proposed two basic firewall layouts and description of their configurations. Characteristics of these layouts are verified by penetration testing.

Klíčová slova

Firewall, bezpečnost sítě, bezpečnostní politika, provoz, relace, vybudované spojení, nevybudované spojení, paket, rámec, směrovač, přepínač, důvěryhodná síť, nedůvěryhodná síť, přístupový seznam, příznak, IP adresa, paketový filtr, stavová inspekce, routovací mód, transparentní mód, demilitarizovaná zóna, útok, penetrační testování.

Keywords

Firewall, network security, security policy, traffic, session, established connection, embryonic connection, packet, frame, router, switch, trusted network, untrusted network, access list, flag, IP address, packet filter, statefull inspection, routed mode, transparent mode, demilitarized zone, attack, penetration testing.

Seznam použitých symbolů a zkratek

ASA (Adaptive Security Appliance) – bezpečnostní zařízení

ASDM (Adaptive Security Device Manager) – grafické rozhraní pro management ASA zařízení

ASIC (Application-specific integrated circuit) – integrovaný obvod

BPDU (Bridge Protocol Data Units) – výměna informací mezi přepínači

CLI (Command-line interface) – textové rozhraní

DoS (Denial of Service) – typ útoku

DNS (Domain Name System) – překlad jmen

FTP (File Transfer Protocol) – protokol pro přenos dat

HTTP (Hypertext Transfer Protocol) – protokol pro výměnu hypertextových dokumentů

ICMP (Internet Control Message Protocol) – protokol IP sady

IDS (Intrusion detection system) – detekce útoků

IOS (Internetwork Operating System) – operační systém společnosti Cisco

IPS (Intrusion prevention system) – prevence útoků

IPX (Internetwork Packet Exchange) – protokolová sada operačních systémů Novell a NetWare

NAT (Network address translation) – překlad adres

OS (Operating system) – operační systém

OSI (Open Systems Interconnection) – standardizace počítačových sítí a protokolů

SMTP (Simple Mail Transfer Protocol) – protokol pro přenos zpráv elektronické pošty

SSH (Secure Shell) – zabezpečený komunikační protokol

TCP/IP (Transmission Control Protocol/Internet Protocol) – protokolová sada

MAC address (Media Access Control address) – fyzická adresa

Obsah

1. Úvod

Zabezpečení sítí umožnilo rozšíření využití internetu v dnešním světě obchodu, zábavy a sdílení informací. Stále více činností spojených s obchodem a službami se přesunuje do oblasti veřejných sítí a poskytovatelé těchto služeb musí zabezpečit jak svá vlastní data, tak soukromé informace svých zákazníků před zneužitím.

Velké škody mohou způsobit různé útoky, např. neautorizovaný přístup jak z veřejné sítě (útok hackera) nebo z vnitřní části sítě (útok nespokojeného pracovníka). Důsledkem takovýchto útoků může být možné zneužití soukromých dat zákazníků, čímž může dojít k negativnímu ovlivnění image společnosti a její pověsti na trhu. Zároveň se může snížit produktivita společnosti a dokonce může dojít k ukončení činnosti společnosti. Minimálně mohou tyto útoky narušit partnerské vztahy se zákazníky a obchodními partnery, kteří se poté mohou dotazovat na schopnost společnosti střežit důvěrné a jim svěřené informace.

Z těchto důvodů je pro společnosti a organizace důležité přijmout nezbytné kroky k zabezpečení své síťové infrastruktury a soukromých dat jak ze strany vnějších útoků, tak ze strany útoků uvnitř organizace.

Zabezpečení celé síťové infrastruktury může být velice náročný úkol. Čím více různých služeb organizace nabízí (např. elektronické bankovnictví, intranet a extranet, emailové služby), tím více nabírá tento úkol na komplexnosti. Z hlediska nákladů na zabezpečení celé síťové infrastruktury (zařízení potřebné pro implementaci této bezpečnostní politiky a kvalifikovaný personál pro jeho obsluhu), musíme vždy zvažovat náklady vydané na toto zabezpečení proti ztrátám, které by mohly vzniknout při porušení této bezpečnostní politiky.

Mnoho společností připojených k internetu jsou chráněny firewally, které jsou navrženy k ochraně vnitřních sítí před útoky, které přicházejí zvenčí, z internetu. Firewally poskytují bariéru proti neautorizovanému přístupu do privatní sítě. Jsou také umístněny uvnitř privátních sítí k prevenci útoků uvnitř sítě.

V této diplomové práci se budu zabývat zabezpečením sítí pomocí firewallu. A to konkrétně bezpečnostní bránami typu ASA, které kromě jiných funkcí nabízí funkci síťového firewallu. ASA (Adaptive Security Appliance) je řešením výrobce firmy Cisco, která je jedna z předních celosvětových dodavatelů síťových technologií.

První část diplomové práce je věnována problematice firewallů, jejich historickému vývoji a rozdělení.

Druhá část se zabývá návrhem a různými možnostmi zabezpečení sítí pomocí firewallů a jejich konfigurací.

Třetí část je věnována ověření vlastností navržených zapojení pomocí penetračních testů.

2. Co je to firewall?

Původně název firewall označoval nehořlavý segment, který odděluje jednotlivé části např. budovy, aby se zabránilo šíření ohně dále budovou . Ve světě síťových technologií je tento termín používán jako metafora pro oddělení vntřní síťové infrastruktury od nebezbečí vznikajících ve vnější části sítě. Firewall nám dovoluje segmentovat naši síť mezi rozdílné fyzické subnety, čímž nám pomáhá omezovat rozšiřování možných škod z jednoho subnetu do dalších.

Funkce firewallu mohou být implementovány jak do specializovaného softwaru, který je nainstalovaný v osobním počítači a nebo je již součástí operačního systému tohoto osobního počítače a poskytuje ochranu uživateli, tzv. osobní firewall, nebo do síťového zařízení, které plní funkci např. routeru a kromě routovaní paketů umožňuje operační systém tohoto zařízení plnit některé funkce firewallu např. filtrovaní provozu s využitím jednoduchých paketových filtrů ACL (Access Control Lists).

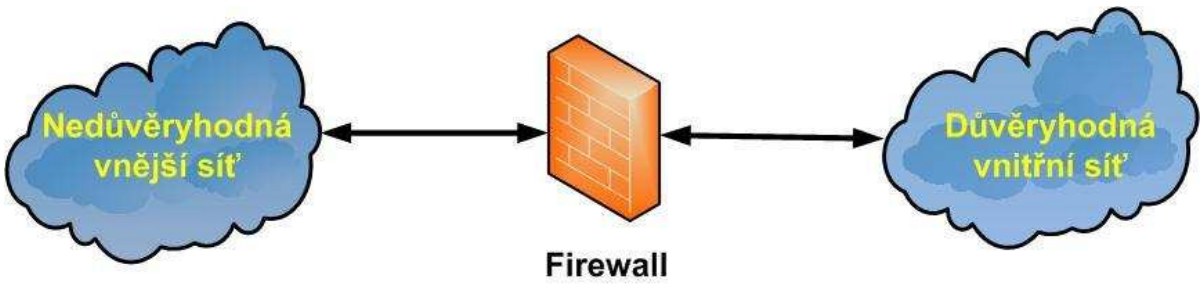
Další skupinou jsou specializovaná bezpečnostní zařízení, tzv. síťové firewally. V těchto případech zařízení využívá integrovaných obvodů pro specifické aplikace ASIC (Application-Specific Integrated Circuit) pro plnění funkcí jak bezpečnostního firewallu, tak zařízení prevence vniknutí IPS (Intrusion Prevention System) nebo vytváření a ukončování IPSec VPN tunelů a atd.

Obecně lze tedy firewally rozdělit do dvou hlavních skupin jak je zřejmé z předchozího popisu a to na osobní a na síťové firewally. Osobní firewally zajišťují bezpečnost pouze jednoho uživatele, pracovní stanice nebo serveru. Síťové firewally mohou zajišťovat bezpečnost i více takovýchto zařízení, rozčleněných např. podle logické nebo fyzické topologie zapojení.

Dalším způsobem může být třídění podle použité technologie, kterou firewall využívá ke své funkci, a to na:

- Paketové filtry (Packet filters)
- Proxy filtry (Application layer firewall, Proxy filters)
- Stavové paketové filtry (Statefull inspection)

Dále pak např. na transparentní firewally, virtuální firewally atd.



Obr. 1: Umístnění firewallu mezi vnější a vnitřní síť.

3. Historie firewallu

Technologie firewallů se objevila ke konci osmdesátých let minulého stoletní, kdy internet byla ještě docela nová technologie z hlediska globálního využití a připojení uživatelů. Předchůdci firewallů z hlediska síťové bezpečnosti byly routery používané v osmdesátých letech minulého století k oddělení jednotlivých sítí. Pohled na Internet jako na relativně malou a bezpečnou komunitu uživatelů, kteří oceňovali otevřenost sdílení informací a spolupráci uživatelů, byl ukončena koncem osmdesátých letech minulého století, kdy se objevily první větší bezpečnostní útoky a narušení.

Následuje stručný chronologický přehled nejdůležitějších technologií firewallů:

3.1. Paketové filtry

Vůbec první pojednaní o firewall technologiích bylo publikováno v roce 1988, kdy inženýři z Digital Equipment Corporation (DEC) vyvinuli filtr známý jako paketový filtr - firewall. Tento

základní systém byl první generací, která nastartovala a rozvinula další vývoj této bezpečnostní technologie. V Bellových laboratořích AT&T, Bill Cheswick a Steve Bellovin pokračovali ve výzkumu paketového filtrování a vyvinuli fungující model postavený na architektuře první generace filtrů.

3.2. Filtry na aplikační vrstvě, proxy filtry

Gene Spafford z Univerzity v Purdue, Bill Cheswick z AT&T laboratoří a Marcus Ranum popsali další generaci firewallů známou jako firewall na aplikační vrstvě. Tato práce vedla k vývoji komerčního produktu, který byl uveden na trh společností DEC pod jménem DEC SEAL. Aplikační firewally poskytli mnohem větší zabezpečení a spolehlivost v porovnání k paketovým filtrům. Aplikační filtry pracují na všech sedmi vrstvách OSI (Open System Interconnection) modelu.

3.3. Stavová inspekce

Nir Zuk a jeho tým ze společnosti Check Point jsou považovány za tvůrce stavové inspekce v polovině devadesátých let minulého století. Před příchodem stavové inspekce byla využívána bezstavová inspekce, která nahlížela na každý příchozí nebo odchozí paket izolovaně. Takový bezstavový firewall neměl mechanizmus, kterým by zjistil, zda příchozí nebo odchozí paket náleží nějaké části již vytvořeného spojení nebo jde o paket nenáležející k žádnému spojení. Stavové inspekce využívají moderní firewally.

4. Základní technologie firewallů

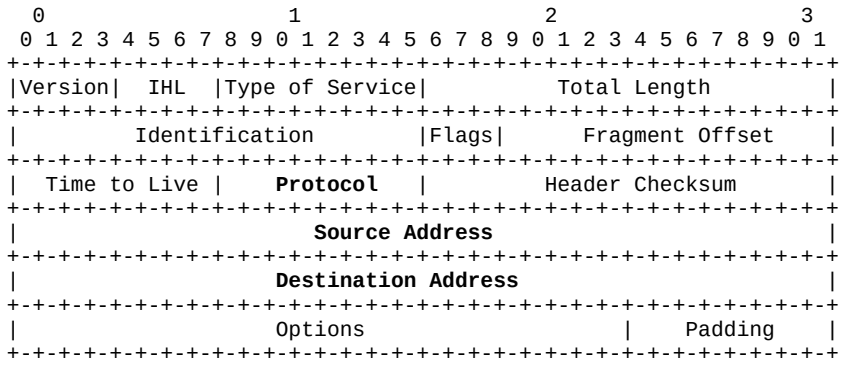
Tato část se zaměří na popis různých technologií, které firewally využívají ke své činnosti a které byly představovány postupem času, jak docházelo k jejich vývoji a zdokonalování .V praxi je běžné že výrobce využívá různých kombinací těchto technologií. Každá z uvedených technologií má totiž své klady a zápory a také různé požadavky na hardwarové a softwarové zdroje zařízení, na kterém běží. Jako např. vytížení CPU (Central Processor Unit), velikost interní flash paměti nebo RAM paměti.

4.1. Paketové filtry

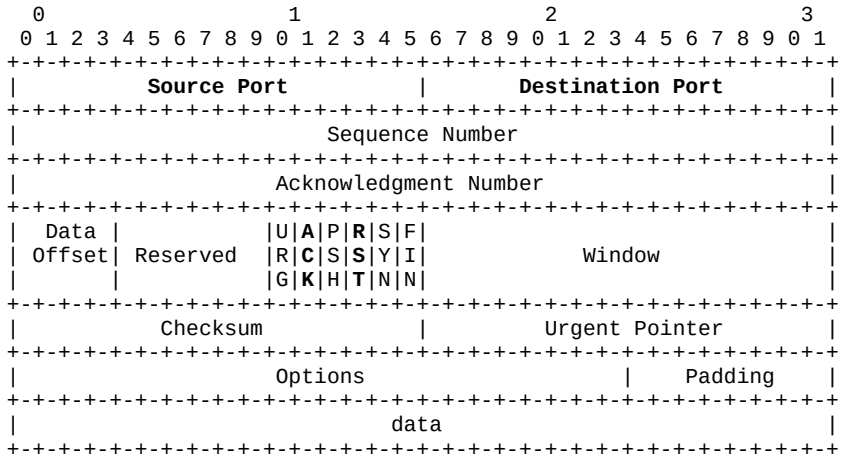
Paketové filtry jsou síťová zařízení která filtrují provoz na základě jednoduchých informací uvnitř TCP/IP paketu. Tato zařízení pracují tak, že si neuchovávají žádné informace ve stavové tabulce, tudíž paketový filtr se rozhoduje podle platných přístupových seznamů (např. Access Lists), zda provoz povolí (permit, accept), nebo zamítne (deny, drop). Po zpracování si ale již žádné další informace neukládá a u dalšího přijatého paketu se celý proces opakuje. Pro obousměrný provoz proto musí být nakonfigurován tak, aby povoloval i pakety přicházející z této druhé strany. Jednoduchým příkladem paketových filtrů jsou access lists v Cisco IOS nebo ipfwadm v linuxových distribucích. Ačkoli tyto filtry poskytují ochranu proti širokému spektru útoků, nejsou dostatečně dynamické na to, aby mohly být považovány za plnohodnotné firewally. Jejich hlavním rysem je limitovat příchozí provoz do chráněné sítě, zatímco odchozí provoz do nechráněné sítě zůstane neomezený.

Při konfiguraci paketového filtru se vytváří pravidla, která definujeme pro zdroj nebo cíl komunikace. Paketový filtr povoluje nebo zakazuje provoz na základě jedné nebo více uvedených hodnot v hlavičce IP paketu nebo čísla portu TCP/UDP, které jsou:

- Zdrojová IP adresa
- Cílová IP adresa
- Protokol – TCP, UDP,NTP, ICMP
- Číslo zdrojového portu
- Číslo cílového portu



Obr. 2: Hlavička IP paketu.



Obr. 3: Hlavička TCP segmentu.

Nevýhody jednoduchých paketových filtrů

- Paketovým filtrem mohou procházet libovolně sestrojené pakety, u kterých pouze stačí aby odpovídaly kritériím ACL
- Pakety mohou projít paketovým filtrem také ve fragmentované podobě
- Obtížnost údržby, implementace a vytváření rozsáhlých ACL
- Některé (hlavně multimediální aplikace) si domlouvají čísla portů na kterých bude aplikace běžet během spojení nebo využívají portů více, což znemožňuje vyzvářet ACL pro takovéto aplikace

4.2. Aplikační firewall - proxy filtry

Proxy filtr inspektuje pakety i na vyšších vrstvách modelu OSI, obvykle pak na vrstvě transportní až aplikační. Aplikační firewally jsou obvykle navrženy tak, aby kontrolovaly jednu aplikaci nebo více specifické aplikace a služby (jako např. databázové nebo webové servery) na rozdíl od stavové inspekce, která může inspektovat téměř jakýkoliv provoz v síti.

Proxy filtr zaručuje důkladnou kontrolu paketů jejich inspekcí na vyšších vrstvách, zároveň ale také díky této vlastnosti snižuje propustnost sítě, která je poté omezena rychlostí zpracování těchto paketů. Koncová zařízení uvnitř zabezpečené sítě poté komunikují v zastoupení proxy serveru s koncovými zařízeními vnější nezabezpečené sítě a naopak.

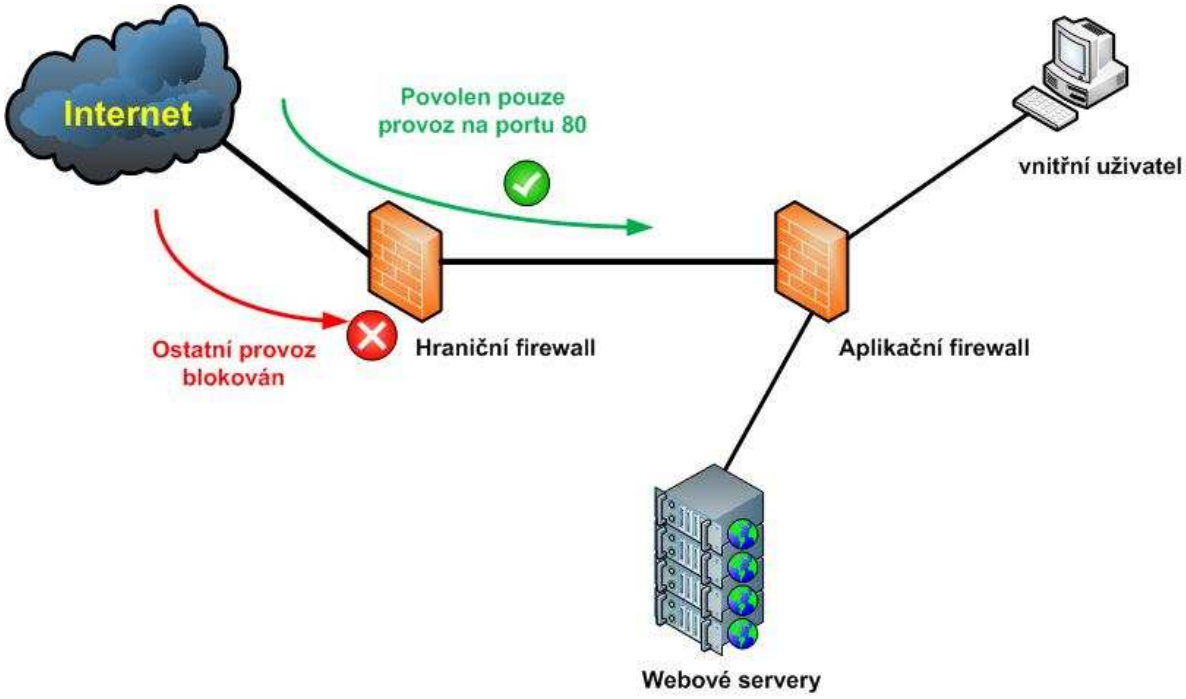
Koncová zařízení musí pro získání přístupu do sítě projít procesem v němž se ustanoví stav relace, proběhne autentizace a uplatní se přístupová práva. Koncová zařízení vnitřní zabezpečené sítě se k vnějším službám připojují přes aplikační programy (proxy), které běží na bráně (proxy server), propojující vnitřní síť s vnější nezabezpečenou sítí.

Jeden ze způsobů činnosti firewallu s proxy filtrem funguje tak, že koncové zařízení na vnitřní zabezpečené síti nejprve vytvoří komunikační relaci vůči samotnému firewallu, poté se koncové zařízení autentizuje a podle jeho identifikačních údajů a hesla mu firewall povolí určitý okruh přístupových práv do dalších sítí. Při tomto způsobu činnosti firewallu se vytvářejí vždy dvě samostatné relace, jedna od koncového zařízení k proxy a druhá od proxy do cílového koncového zařízení, kam požadovaný provoz směřuje.

Aplikační firewally mohou pracovat také jako podpora hraničních firewallů, které jsou umístněny na hranici vnitřní a vnější nedůvěryhodné sítě (tzv. perimeter firewall).

Útoky proti webovým aplikacím jsou často vedeny s využitím služeb HTTP port 80 a SSL na portu 443, kde se přes povolený provoz na těchto portech vyhledávají možné zranitelnosti na cílovém serveru. Úplné zablokování veškerého provozu na těchto portech není možné, pokud chceme provozovat veřejně dostupný webový server. Zároveň IDS (Intrusion Detection System) a IPS (Intrusion Prevention System) systémy běžně inspektují pakety s využitím signatur pro již známé útoky a nejsou vždy efektivní proti všem možným hrozbám.

Na obrázku Obr. 4 je znázorněno doporučené umístnění aplikačního firewallu. Hraniční firewall na základě své bezpečnostní politiky povoluje provoz, který je v tomto případě na portu 80 směřovaný k webovému serveru a prochází ještě aplikačním firewallem, na kterém se provádí další inspekce na aplikační vrstvě. Všechen provoz k a od webového serveru je inspektován tímto firewallem.



Obr. 4: Síť s aplikačním firewallem.

Nevýhody proxy filtru:

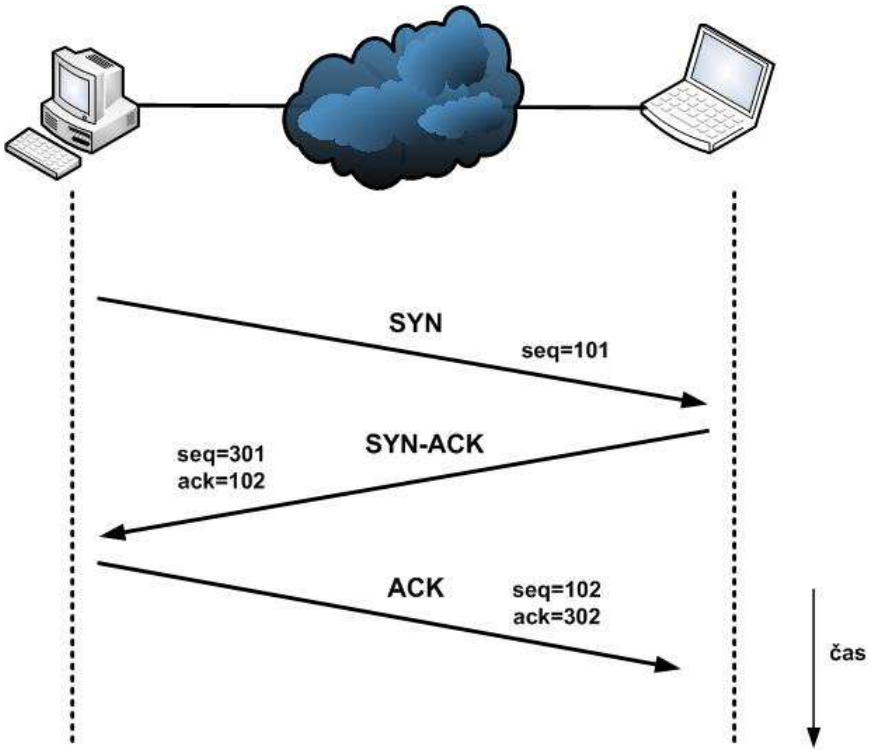
- Náročnost při doplňování nových služeb
- Při zatížení pracuje firewall pomaleji
- Proxy firewall bývá postaven na univerzálním operačním systému, protože pro svoji činnost vyžaduje určité služby operačního systému. Tím vznikají dvě třídy problémů - vyšší zbytečná režie a tedy pomalejší chod firewallu a dále zranitelnost dobře známého operačního systému vůči útokům.

4.3. Stavová inspekce

Tento typ firewallů v sobě spojuje vlastnosti z technologií paketových filtrů a proxy filtrování. Pro každou relaci, která je vedena přes firewall, si uchovává kompletní stavové informace. Při každém navázání nového spojení, ať z vnitřní zabezpečené sítě nebo z vnější nedůvěryhodné sítě, si do stavové tabulky relačních toků ukládá příslušné údaje.

Stavová tabulka relačních toků zahrnuje u každého spojení TCP/UDP, přidruženého příslušné komunikační relaci, informace o zdrojové a cílové IP adrese, čísla portů, údaje o pořadových číslech TCP, a případné doplňující příznaky. Při navázání relace přes firewall se vytvoří objekt spojení, veškeré pakety se poté porovnávají s relačním tokem, který je zaznamenaný ve stavové tabulce relačních toků, a průchod firewallem se povolí jen v případě, že k nim existuje příslušné spojení.

Stavová inspekce závisí na třícestném procesu navázání komunikace, který se využívá u protokolu TCP (tzv. three-way handshake). Pokud je při komunikaci využit protokol UDP, stavová inspekce nespoléhá na mechanizmy, které se využívají u TCP. Při žádosti o sestavení spojení je koncovým zařízením poslán paket, který má nastavený příznak (flag) v hlavičce TCP datagramu na hodnotu, která vyjadřuje příznak SYN (synchronization). Všechny pakety s hodnotou příznaku nastavenou jako SYN jsou firewallem považovány za nová spojení. Jestliže je cílové zařízení dostupné, odpovědí bude paket s příznakem ve kterém je nastaven také bit ACK (acknowledge), SYN ACK. Odpovědí na SYN ACK je paket s nastavenou hodnotou příznaku ACK. Toto spojení je poté považováno za vybudované (ESTABLISHED). Firewall poté bude přeposílat všechny odchozí pakety s vnitřní sítě a povolovat pouze příchozí pakety s vnější nedůvěryhodné sítě jestliže náleží do již některého z vybudovaných spojení. Tato ochrana zajišťuje že žádné pakety z nevyžádaného spojení nebudou přeposílány zařízení na vnitřní zabezpečené síti.



Obr. 5: Navázaní komunikace u TCP protokolu.

Jako prevence před zaplněním stavové tabulky se odstraní relace, které nejsou aktivní a neprochází jimi žádný provoz za určitou stanovenou dobu. Mnoho aplikací, proto aby se vyhnulo odstranění ze stavové tabulky a tím i ukončení vybudovaného spojení přes firewall z důvodů neaktivity komunikujících koncových zařízení, posílá tzv. keepalive zprávy pro udržení spojení. I samotné firewally mohou být nastaveny tak, aby tyto zprávy komunikujícím aplikacím zasílaly. Mnoho firewallů jsou schopny udržovat informace i o nespojově orientovaných protokolech, jako je UDP. Takové relace se považují za vybudované (ESTABLISHED) po prvním paketu, který projde firewallem. Tyto relace jsou ukončovány vypršením času platnosti relace.

Stavová inspekce monitorováním jednotlivých relací poskytuje zvýšenou efektivitu při inspekci paketů při průchodu paketu firewallem, kdy se pouze kontroluje stavová tabulka a ne všechna pravidla nastavená na firewallu jako u paketových filtrů (která mohou být dost rozsáhlá). Další výhodou může také být, že při obnově pravidel firewallu se stará stavová tabulka odstraní a vytvoří se nová podle aktuálně nastavených pravidel.

5. Zařízení ASA

Následuje popis zařízení ASA 5505, které použijeme ve funkci firewallu k zabezpečení sítě v kapitole č. 7 .

Ke konfiguraci firewallu lze využít:

- ASDM (Adaptive Security Device Manager)

  Grafické rozhraní pro práci s ASA zařízeními

- CLI (Command Line Interface).

  Příkazový řádek

Ke konfiguracím budeme pouze využívat CLI, které je pro pochopení vlastností firewallu vhodnější.

Obr. 6: ASA 5505.

Obr. 7: ASA 5505.

5.1. Technická specifikace ASA 5505

ASA 5505 byla představena v roce 2006, je vhodná pro malé podniky nebo pobočky a pracovníky se vzdáleným přístupem k podnikovým zdrojům.

Hardwarové parametry

- Typ CPU - AMD Geode LX
- Rychlost CPU - 500 MHz
- Chipset Geode CS5536
- Standardní velikost paměti RAM - 256 MB
- Bootovací Flash - ATA CompactFlash
- Standardní velikost paměti Flash - 64MB
- Maximální počet virtuálních rozhraní - 3
- Síťový chipset - Marvell 88E6095
- Počet fyzických rozhraní: 8 x 10/100 Mbit, 2 x PoE (Power over Ethernet)

- Počet logických rozhraní:
    - Base licence - 3 (bez využití trunků),
    - Security Plus licence – 20 (s využitím trunků)

Výkonové parametry

- Maximální počet spojení s firewallem/sec. - 4000
- Propustnost (cleartext) - 150 Mbit/s
- Propustnost AES/Triple DES - 100 Mbit/s
- Maximální počet současně probíhajících spojení – 10 000
- Maximální počet site-to-site a remote access VPN relací - 10
- Maximální počet SSL VPN uživatelských relací - 25
- Maximální počet paketů za vteřinu (při velikosti 64 byte) - 85 000

Další parametry

- Minimální verze OS - 7.2.1
- Podpora rozšířitelných modulů AIP-SSC
- Podpora SSL VPN
- Podpora security context (virtualizace firewallu) – Ne
- GUI (Graphical User Interface) grafické uživatelské rozhraní – ASDM
- CLI (Command Line Interface) rozhraní příkazové řádky – konzole, telnet, SSH (Secure shell)
- AAA (Authorization, Authentization, Accounting) , Cut-through proxy – Ano
- Routování – statické routy, RIP, EIGRP, OSPF

5.2. Verze ASA

Verzi operačního systému použitou licenci a jiné systémové informace z ASA zařízení lze zjistit příkazem *show version* .

Naše použitá verze OS je 7.2(4) a obsahuje Base licence.

```
ASA# show version

Cisco Adaptive Security Appliance Software Version 7.2(4)
Device Manager Version 5.2(4)
Compiled on Sun 06-Apr-08 13:39 by builders
System image file is "disk0:/asa724-k8.bin"
Config file at boot was "startup-config"

Hardware:   ASA5505, 256 MB RAM, CPU Geode 500 MHz
Internal ATA Compact Flash, 128MB
BIOS Flash M50FW080 @ 0xffe00000, 1024KB

Licensed features for this platform:
Maximum Physical Interfaces : 8
VLANs                       : 3, DMZ Restricted
Inside Hosts                : 10
Failover                    : Disabled
VPN-DES                     : Enabled
VPN-3DES-AES                : Enabled
VPN Peers                   : 10
WebVPN Peers                : 2
```

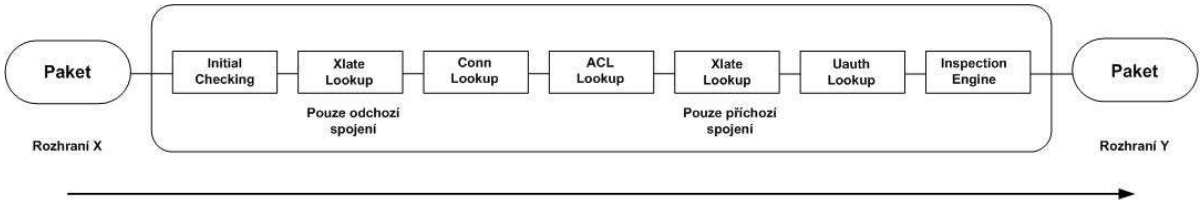
```
Dual ISPs                      : Disabled

VLAN Trunk Ports               : 0


This platform has a Base license.
```

5.3. Logika ASA zařízení

ASA zařízení (Adaptive Security Appliance) je samostaný hardwarový box, který nabízí kromě funkce firewallu také funkce IPS (Intrusion Prevention System) a VPN (Virtual Private Networks). Dále se zaměříme pouze na funkci firewallu ASA zařízení.

ASA firewall považujeme za hybridní systém, protože využívá technologie paketového filtru, proxy filtru i stavové inspekce. Hlavní částí ASA zařízení je Adaptive Security Algorithm (ASA). ASA algoritmus je součást proprietárního operačního systému ASA firewallů. Zajišťuje stavovou inspekci paketů a uchovává relační toky, tvořené podle zdrojové a cílové adresy, které se zaznamenávají do stavové tabulky relačních toků.

Inspekce provozu u ASA firewallu je dána postupným vykonáváním funkcí, na obrázku obr. xx je zobrazen tento postup, kdy paket dorazí na rozhraní X a vystupuje rozhraním Y. Následující popis bude věnován jednotlivým fazím inspekce.



Obr. 8: Pořadí vykonávání jednotlivých funkcí ASA firewallu.

5.3.1. Vstupní kontrola - Initial Checking

Na vstupním rozhraní X je kontrolována integrita přijatého paketu, jedna z nejdůležitějších informací v hlavičce paketu je zdrojová IP adresa. Při normálním provozu hledá firewall vhodnou routu pro cílovou IP adresu přijatého paketu a podle ní najde příslušné výstupní rozhraní. Zdrojová adresa zůstává bez kontroly do doby kdy cílový host posílá odpověď. Host může záměrně podvrhnout nepravou zdrojovou IP adresu v paketu (tzv. address spoofing) a vydávat se za jiného hosta na síti. Adress spoofingu lze využít při DoS útocích. RFC 2827 popisuje metodu, kterou může firewall použít, aby mohl detekovat adress spoofing.

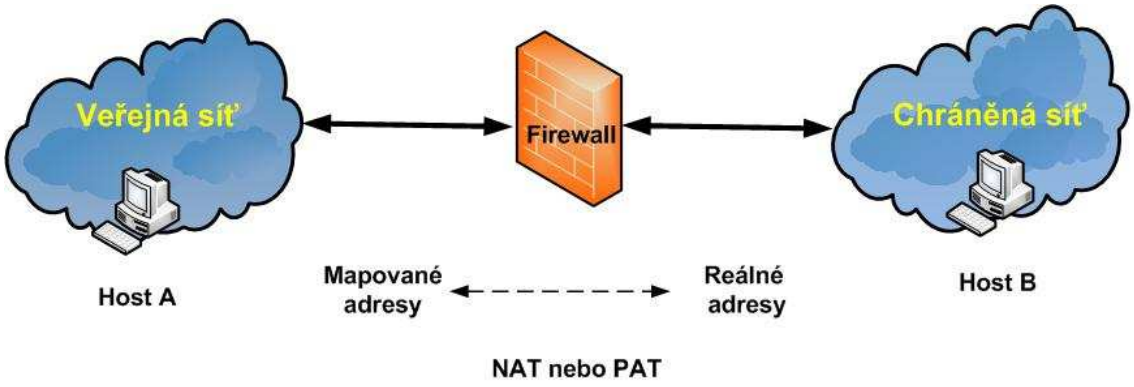
ASA firewally využívají této definované metody ve funkci - RPF (Unicast Reverse Path Forwarding). Pokud je tato funkce na rozhraní firewallu zapnutá, je každá zdrojová IP adresa v příchozím paketu inspektována. Firewall musí ve své routovací tabulce najít zdrojovou adresu přijatého paketu a tato adresa musí zároveň korespondovat s adresami sítí připojených k rozhraní na kterém byl tento paket přijat. Jinými slovy, firewall ověřuje zda paket půjde stejnou cestou zpět při zpětném provozu, kdy cílem komunikace bude tato zdrojová IP adresa. Firewall zahodí všechny pakety, které neprojdou RPF tetstem.

Speciálním případem je rozhraní firewallu, za kterým je veřejná síť. Pokud má firewall defaultní routu asociovanou s tímto rozhraním, je každý paket přijatý na tomto rozhraní, pokud jeho adresa není nalezena v jeho routovací tabulce, RPF testem povolen. Ale pokud by byla použita adresa asociovaná s jiným rozhraním firewallu, RPF test by tento paket zahodil.

Firewall detekuje pouze address spoofing mezi dvěma rozhraními, ví, že použitá adresa již existuje za jiným rozhraním a proto může přijatý paket zahodit. Jestliže host za rozhraním firewallu k veřejné síti využije adresu jiného hosta z veřejné sítě, firewall nemůže tento address spoofing detekovat, protože proběhl pouze za jedním rozhraním firewallu.

5.3.2. Náhled do tabulky překladů - Xlate Lookup

ASA firewall si udržuje tabulku překladů (translation table) nebo xlate tabulku pro každého hosta v chráněné síti, který se účástní některého z právě probíhajících spojení. Překlad může být nastaven staticky nebo dynamicky. Záznamy v xlate tabulce se vytváří až při aktivním spojení.



Obr. 9: Překlad adres.

Host A ve veřejné síti má registrovanou veřejnou IP adresu, host B v chráněné síti má vnitřní IP adresu, která se nazývá skutečná nebo lokální IP adresa. Adresa vnitřního hosta ja přeložena xlate záznamem tak, že vystupuje tato lokální (reálná) adresa do veřejné sítě jako mapovaná nebo globální IP adresa.

Každý záznam v xlate tabulce je uchováván s těmito parametry:

- Použitý protokol (ICMP, UDP, or TCP)
- Lokalní a globalní rozhraní
- Lokalní a globalní IP adresy
- Lokalní a globalní čísla portů
- Příznak - flag (typ xlate)
- Časovač nečinnosti spojení (zvyšuje svoji hodnotu pokud není xlate záznam používán)
- Absolutní časovač (zvyšuje svoji hodnotu od vytvoření xlate záznamu)
- Uauth vazby (pokud je použita autentizace uživatele)

Náhledy do xlate tabulky se provádějí v závislosti na směru provozu.

Pro odchozí spojení (iniciované z vnitřní chráněné sítě) se vytváří xlate záznam v počáteční fázi inspekce, a to z důvodu, že se využívají přeložené globální adresy k vytvoření záznamů o spojeních.

Pro příchozí spojení (iniciované z veřejné nechráněné sítě) se vyhledává záznam v xlate tabulce později, po kontrole nepřeložených globálních adres v ACL.

Firewall využívá xlate záznamy:

- K řízení počtu aktivních spojení, která mohou používat jeden xlate záznam, defaultně – neomezený počet, nebo počet nastavený limitem
- K řízení počtu ne úplně vybudovaných spojení – tzv. embryonic, která mohou používat jeden xlate záznam, defaultně – neomezený počet, nebo počet nastavený limitem
- List aktivních spojení, záznamům vyprší platnost a je smazán z xlate tabulky, pokud není aktivní po danou časovou periodu

5.3.3.Náhled do tabulky spojení - Conn Lookup

Každé spojení, které se snaží o průchod ASA firewallem, se zkoumá a udržují se o něm aktuální informace v tabulce spojení (connection table, conn table). To se nazývá stavová inspekce. Pokud je povoleno navázání spojení (ACL povolují tento provoz), je vytvořen záznam v conn tabulce. Stav spojení a chování paketů přichazejích jak ze zdrojové nebo cílové IP adresy, musí dodržovat pravidla protokolové sady TCP/IP. Jakákoliv odchylka od tohoto chování způsobí vynucené ukončení spojení.

Každý záznam v conn tabulce je uchováván s těmito parametry:

- Použitý protokol (ICMP, UDP, or TCP)
- Lokální a cílové IP adresy (jsou zde použity stále lokální adresy i po náhledu do xlate tabulky)
- Lokalní a cílové čísla portů
- Příznaky (flag) pro opravu typu a stavu spojení
- Časovač nečinnosti spojení (zvyšuje svoji hodnotu, pokud pakety neprocházejí spojením)
- Bitový čítač (celkový objem dat přenesených daným spojením)
- Lokální a cílové TCP sekvenční čísla (seq number)

Záznamy v conn tabulce jsou odstraněny po určité časové periodě kdy spojením neprochází žádné pakety.

Záznamy jsou také odstraňovány po určité krátké časové periodě, kdy nedojde k úplnému vybudování TCP spojení.

Pokud je při spojení využíván TCP protokol, ASA firewall dokáže generovat náhodná sekvenční čísla (ISN - random initial sequence number), která se využívají při vytváření spojení směrem k hostům v nechráněné síti. Některé aplikace nejsou schopny generovat opravdu náhodné sekvenční čísla, což může vést k předpovídatelnosti těchto sekvenčních čísel a využití této slabiny k nabourání se do spojení (tzv. session hijacking). Firewall nahradí při vytváření TCP spojení sekvenční číslo za opravdu náhodné ISN, které snižuje riziko session hijackingu a je plně transparentní pro lokálního i cílového hosta.

5.3.4. Náhled do přístupových seznamů - ACL Lookup

Před tím, než se spojení může vybudovat, musí být procházející pakety povoleny ACL. Na firewallu můžeme nakonfigurovat mnoho ACL, ale vždy pouze jeden může být aplikován na rozhraní firewallu ve specifickém směru (ven z rozhraní – out, nebo dovnitř do rozhraní – in). ACL nejsou použity pro kontrolu stavu spojení, povolí nebo jen zamítnou pakety přicházející v jednom směru. Ve směru, ve kterém se inicializuje sestavní spojení.

Pro bezspojové protokoly jako je ICMP (Internet Control Message Protocol), ACL zamítnou nebo povolí všechny pakety ve směru ve kterém jsou aplikovány.

Standardně nejsou žádné ACL konfigurovány a aplikovány na rozhraních ASA firewallu. Požadavek na vytvoření spojení je povolen pouze z rozhraní s vyšší úrovní zabezpečení na rozhraní s nižší úrovní zabezečení. Aplikací ACL na rozhraních s nižší úrovní zabezpečení umožníme vytvářet i spojení z nižších úrovní zabezpečení do vyšších.

5.3.5. Náhled do uživatelských autentizací - Uauth Lookup

ASA firewall může také autentizovat komunikující uživatelé při vytváření spojení. Po úspěšné autentizaci si firewall udržuje ověřené uživatelské údaje a schvalování dalších požadavků na nové spojení je urychleno. Firewall tak funguje jako autentizační proxy a už jednou ověřená identita uživatele se nemusí vícekrát ověřovat.

Uživatelská autentizace probíhá mezi firewallem a AAA (Authentication, Authorization, Accounting) serverem, jako např. RADIUS (Remote Authentication Dial-In User Service) nebo TACACS+ (Terminal Access Controller Access Control System Plus) serverem.

Po úspěšné autentizaci uživatele může firewall také ověřovat autorizaci daného uživatele přes AAA server. Tyto informace slouží k tomu aby se uživatel dostal pouze ke zdrojům, ke kterým má mít přístup při komunikaci přes firewall.

Firewall si pro vykonávání těchto funkcí udržuje tabulku autentizovaných uživatelů ve které jsou záznamy pro jednotlivé autentizované uživatele a jejich vlastnosti (uauth). Záznam uath obsahuje zdrojovou IP adresu, autorizační ACL (pokud existuje) a hodnotu časovače relace.

Pokud je uživatel autentizován, může vytvářet nová spojení do doby, než vyprší absolutní uauth časovače. Další časovač hlídá uživatelovu aktivitu, jestli přijímá nebo posílá data a pokud vyprší tento časovač, záznam uath je z autentizační tabulky smazán a všechna spojení, vztahující se k tomuto uživateli, jsou ukončena.

5.3.6.Mechanizmus inspekce - Inspection Engine

ASA firewall inspektuje každé spojení a aplikuje pravidla podle použitého protokolu. Tento proces byl nazýván jako fixup nebo inspekce na aplikační vrstvě. Inspekce probíha jak pro nespojově orientované protokoly, jako ICMP nebo UDP, tak pro spojově orientované jako TCP.

5.3.6.1. Inspekce ICMP

ICMP je nespojově orientovaný protokol, proto nemůže firewall sledovat stavové informace o ICMP mezi dvěma hosty. Firewall proto využívá při inspekci tohoto protokolu na tabulku překladů xlate a přístupové seznamy ACL, žádný záznam conn nebude vytvořen v tabulce spojení. Firewall povoluje pouze jednu odpověď ICMP reply na ICMP request, která jím prošla.

5.3.6.2. Inspekce UDP

UDP je také nespojově orientovaný protokol, host může odeslat paket jinému hostu bez očekávání jakékoliv odpovědi, jako např. u RTP (Real-Time Transport Protocol) pro přenos provozu, který přenáší hlas. Ale například DNS (Domain Name System) využívá UDP protokolu pro oboustranou výměnu informací mezi hosty bez nutnosti vybudování spojení.

5.3.6.3. Inspekce TCP

TCP je spojově orientovaný protokol, firewall může sledovat přesný stav výměny informací. U každého TCP spojení firewall zkoumá zdrojovou a cílovou adresu, navzájem použité porty, sekvenční čísla , hodnoty potvrzení ACK (acknowledgment) a příznaky TCP. Pakety které mají neočekávané hodnoty, nejsou částmi vytvořeného spojení a jsou zahozeny. TCP spojení je inspektováno záznamy v xlate tabulce, ACL a stavové tabulce. Záznamy ve stavové tabulce mají také své příznaky, které reflektují aktuální stav TCP spojení. Například stav trojcestného navázání spojení (three-way handshake) při iniciaci spojení je označen příznakem, který indikuje, který host poslal první SYN bit a od kterého hosta je očekávána odpověď se SYN ACK bitem. Také při ukončení spojení je sledována výměna FIN (finish) bitu.

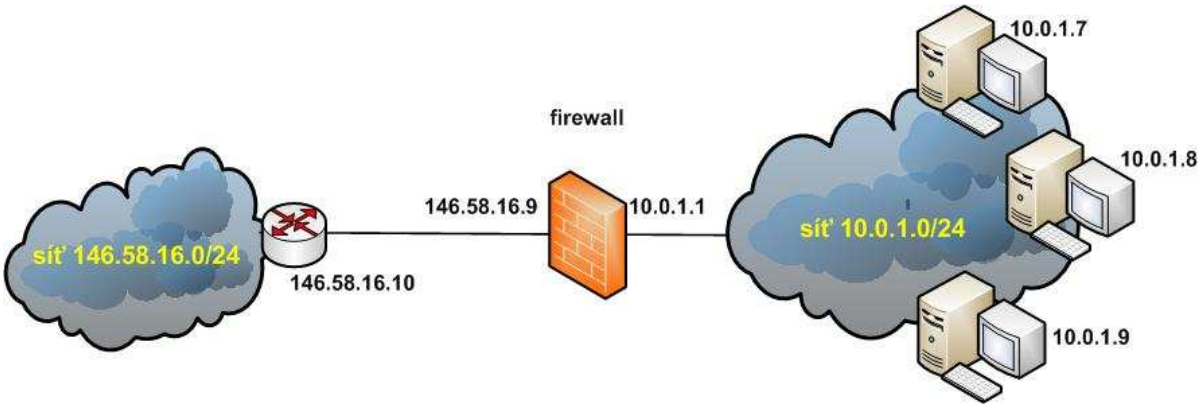
TCP spojení může být ukončeno několika způsoby:

- Oba hosté si mohou navzájem vyměnit pakety s nastavenými FIN bity, firewall sleduje tuto výměnu, spojení je ukončeno správně
- Jeden host pošle druhému RST (reset) bit, čímž žádá druhého hosta o okamžité ukončení spojení
- Firewall udržuje časovač pro každé spojení. Pokud po určitou dobu neprojde žádný paket vytvořeným spojením je spojení ukončeno smazáním záznamu ze stavové tabulky, standardně nastavný čas je jedna hodina.

6. Obecná konstrukce sítě s firewallem

Na následujících schématech si představíme základní zapojení firewallu v datových sítích se kterými se můžeme běžně setkat v praxi.

6.1. Routující mód - routed mode

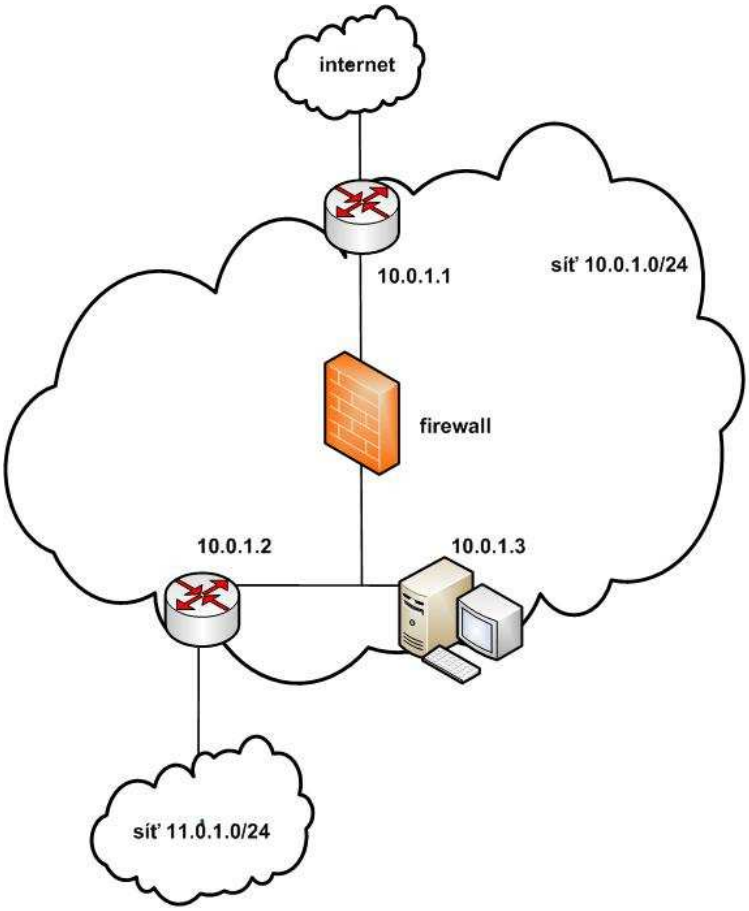


Obr. 10: Firewall v routovací módu.

Firewall v routovacím módu rozděluje síť na nejméně dvě odlišně adresované sítě. V tomto případě je jedno rozhraní firewallu v síti 146.58.16.10/24 a druhé v síti 10.0.1.0/24. Odděluje tak vnější nedůvěryhodnou síť 146.58.16.10 od vnitřní důvěryhodné sítě 10.0.1.0. Pro koncová zařízení uvnitř sítě 10.0.1.0 je defaultní odchozí branou (default gateway) jeho rozhraní 10.0.1.1.

Výhodou tohoto zapojení je, že při aktivním překladu síťových adres na firewallu (Network Adress Translation - NAT) je před uživateli vnější sítě, adresovaní koncových zařízení uvnitř vnitřní sítě skryto. To výrazným způsobem snižuje možnost útoků na vnitřní chráněnou síť.

6.2. Transparentní mód - transparent mode

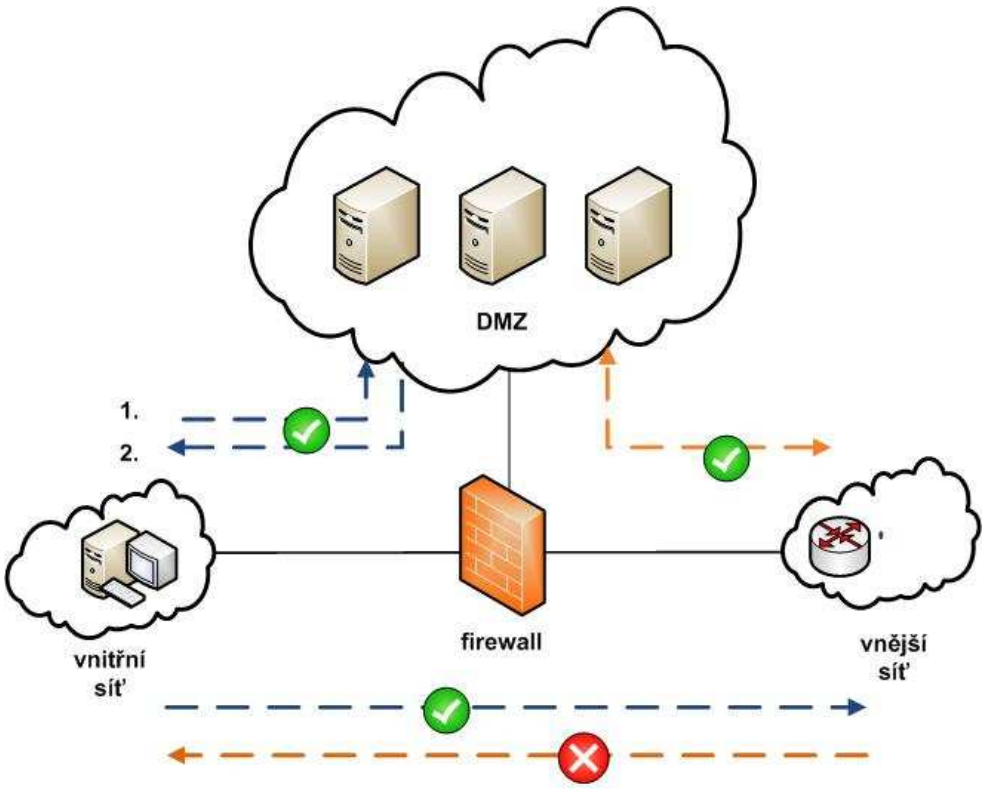


Obr. 11: Firewall v transparentním módu.

Firewall v transparentním módu pracuje tak že všechny jeho rozhraní jsou ve stejné síti, v našem případě 10.0.1.0/24. Defaultní odchozí branou do internetu, pro hosta s adresou 10.0.1.3, je potom rozhraní routeru 10.0.1.1. Firewall pracuje na druhé vrstvě modelu OSI jako přepínač (switch) nebo síťový most (bridge).

Výhodou tohoto zapojení je, že firewall je neviditelný (není „network hop“) pro pakety na třetí vrstvě modelu OSI, které procházejí firewallem. Uživatelé komunikující přes tento firewall nevědí o jeho existenci v síti (viz penetrační testování transparentního firewallu). Další výhodou je že firewallem může procházet jiný než TCP/IP provoz, např. IPX, MPLS, BPDU nebo firewallem mohou procházet multicastové streamy.

6.3. Demilitarizovaná zóna - DMZ

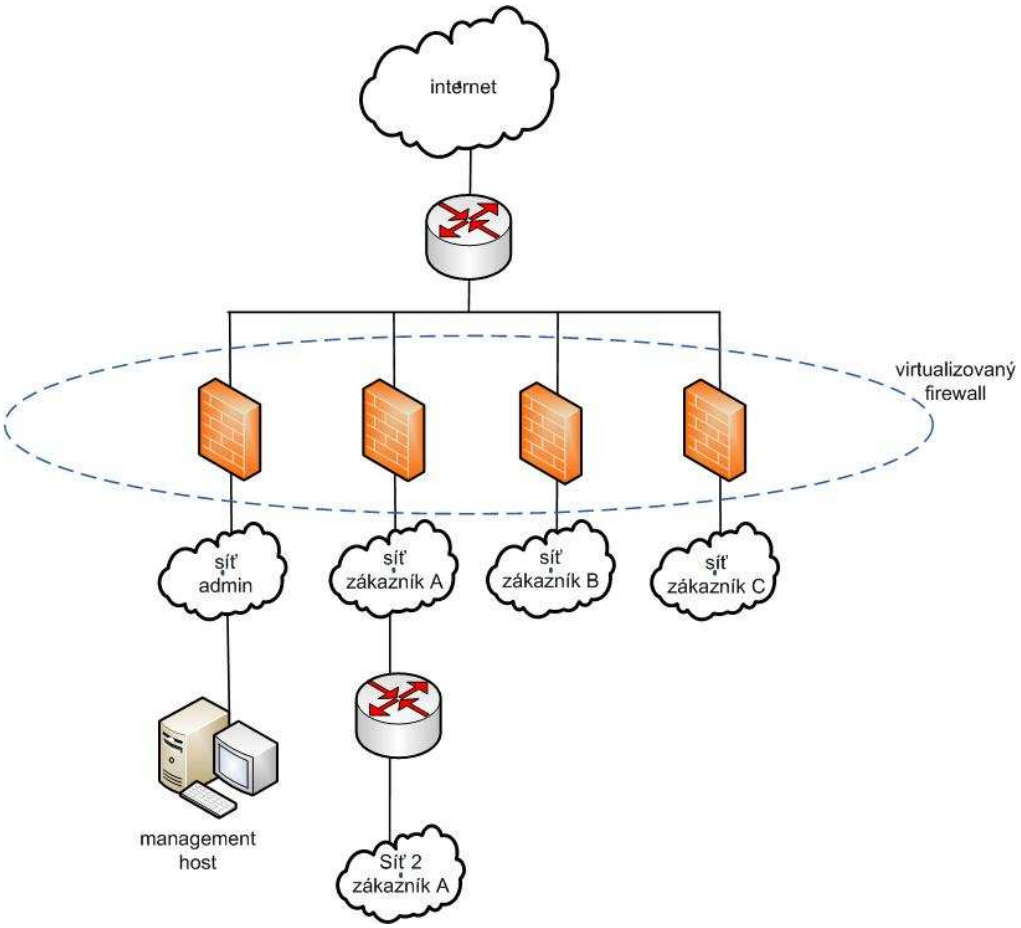


Obr. 12: Zapojení firewallu s DMZ.

V zapojení firewallu s demilitarizovanou zónou je kromě vnitřní a vnější sítě k firewalu připojena i síť, která má označení jako demilitarizovaná zóna (DMZ). V té jsou umístněny veřejně dostupné servery např. webové nebo poštovní servery, na které mají přístup všichni uživatelé z vnější sítě (z internetu). Přístup do DMZ mají také uživatelé vnitřní sítě. Firewall ale zařízením v DMZ nedovolí vytvářet spojení do vnitřní sítě bez toho aby toto spojení nebylo iniciováno z vnitřní sítě. Tím znemožňuje případným útočníkům využít zařízení v DMZ k útoku na zařízení ve vnitřní síti.

Výhodou zapojení s DMZ je to že vytváříme část sítě ve které můžeme umístit zařízení které bychom jinak museli umístit do vnitřní sítě a povolovat k nim přístup z vnější sítě. Tím bychom vytvářeli bezpečnostní díry v pravidlech firewallu. V praxi se běžně využívá i více firewallů různých výrobců k vytváření DMZ .

6.4. Security context - virtualizace firewallu



Obr. 13: Virtualizace firewallu.

Virtualizací firewallu vytváříme několik vzájemně oddělených firewallů, které jsou ve skutečnosti všechny součástí jednoho hardwarového firewallu. To ovšem vyžaduje vyšší nároky na použitý hardware ve firewallu, proto je virtualizace firewallů dostupná pouze u vyšších výrobních řad firewallů (ASA 5510, ASA 5520, ASA 5580).

Výhodou je, že můžeme poskytovat zabezpečení sítě na jednom firewallu současně několika zákazníkům, kteří ale nemají žádné informace o sítích ostatních zákazníků. To přináší finanční úspory, kdy nemusíme nakupovat pro každého zákazníka samostatný hardware. Také administrace takového firewallu je centralizovaná a snažší.

7. Návrh zabezpečené sítě s firewallem

V této části navrhneme a nakonfigurujeme dvě navzájem různá zapojení pro zabezpečení sítě s firewallem ASA 5505. První bude zapojení firewallu v transparentním módu a druhé bude zapojení firewallu v routovacím módu.

7.1. ASA 5505 v transparentním módu

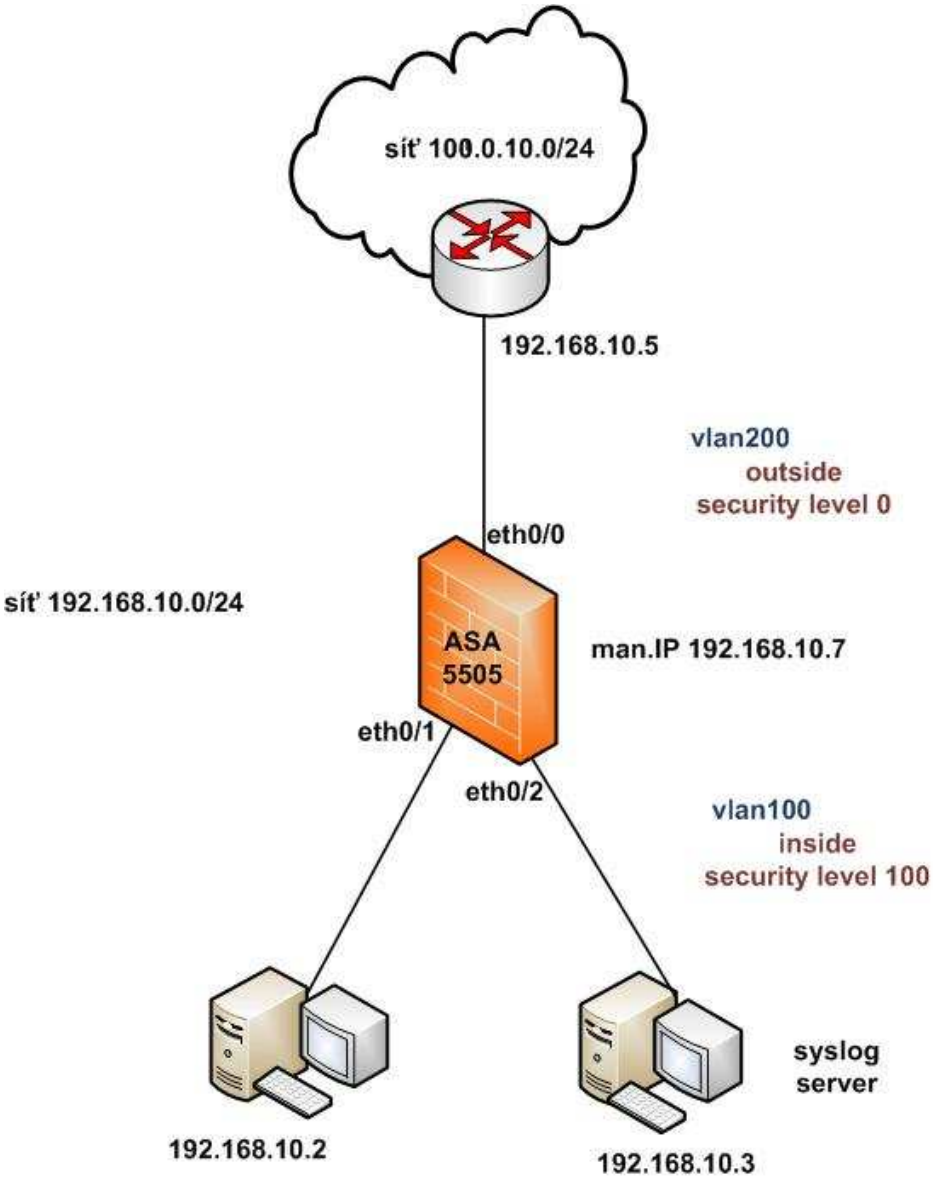
7.1.1. Popis zapojení ASA 5505 v transparentním módu

V tomto zapojení bude firewall součást jedné sítě 192.168.10.0/24, rozhraní firewallu budou na straně důvěryhodné části sítě (inside), nebo nedůvěryhodné části sítě (outside). Vytvoříme v síti jeden logovací server, na který budou posílany systémové logy z firewallu. Umožníme SSH přístup na management IP adresu firewallu pro vzdáleného administrátora. Povolen bude příchozí ICMP provoz ze sítě 192.168.10.0/24 kamkoliv do důvěryhodné části sítě.

Na hosta 192.168.10.2 umožníme přístup z nedůvěryhodné části pouze protokoly pro SMTP, HTTP, DNS a MDNS.

Na hosta 192.168.10.3 bude všechen příchozí provoz z nedůvěryhodné části, s vyjímkou ICMP (ze sítě 192.168.10.0), zakázan.

Nakonfigurujeme inspekci protokolu ICMP, uživatel důvěryhodné části sítě se bude schopen dotazovat na dostupnost uzlu pomocí ICMP protokolu i v jiných sítích, ne pouze 192.168.10.0. Omezíme možnost uživatelů důvěryhodné části sítě vytvářet spojení na námi zakázané webové stránky (xchat a onlinehry).



Obr. 14: Zapojení ASA 5505 v transparentním módu.

7.1.2.Konfigurace ASA 5505 v transparentním módu

```
firewall transparent

hostname FW

domain-name mydomain.cz

enable password EnA8Le
```

V prvním kroku nastavíme mód funkce firewallu příkazem *firewall transparent* (transparentní mód), jméno hosta příkazem *hostname jméno_hosta* (v našem případě FW), doménové jméno příkazem *domain-name jméno_domény* a heslo do enable režimu příkazem *enable password naše_heslo* .

```
interface Vlan100

nameif inside

security-level 100

interface Vlan200

nameif outside

security-level 0
```

Dále vytvoříme rozhraní *Vlan100* a *Vlan200*, vytvořeným vlan rozhraním přiřadíme jméno příkazem *nameif* , tímto jménem se budeme při dalších konfiguracích odkazovat na tato již vytvořená rozhraní. Příkazem *security-level* a číslem od 0 do 100 nastavíme rozhraním jejich důvěryhodnost, kdy *security-level 0*, je rozhraní vnější nezabezpečené sítě a *security-level 100* je vnitřní zabezpečená síť.

```
interface Ethernet0/0

switchport access vlan 200

no shutdown
```

```
interface Ethernet0/1

switchport access vlan 100

no shutdown

interface Ethernet0/2

switchport access vlan 100

no shutdown
```

Těmito příkazy konfigurujeme jednotlivá fyzická rozhraní firewallu (ASA používá označení od *ethernet 0/0* do *ethernet 0/7*) a jejich příslušnost do jednotlivých vlan příkazem *switchport access jméno_vlanu* Vždy je nutno rozhraní aktivovat příkazem *no shutdown*.

```
logging enable

logging trap debugging

logging host inside 192.168.10.3
```

Příkazy *logging enable* povolíme logování na firewallu, *logging host inside 192.168.10.3* posílání systémových logů na hosta 192.168.10.3 (náš syslog server), který je za rozhraním *inside*. Nastavíme nejvyšší úroveň posílaných zpráv (trapů) na hodnotu *debbugging* (7. úroveň).

Možnost zvolit od úrovně číslo 0 (*emergencies*) do úrovně číslo 7 (*debugging*), kdy *emergencies* je nejnižší úroveň generovaných zpráv.

```
access-list ACL_OUT extended permit icmp 192.168.10.0 255.255.255.0 any log

access-list ACL_OUT extended permit tcp any host 192.168.10.2 eq www log

access-list ACL_OUT extended permit udp any host 192.168.10.2 eq domain log

access-list ACL_OUT extended permit tcp any host 192.168.10.2 eq smtp log

access-list ACL_OUT extended permit tcp any host 192.168.10.2 eq 5353 log
```

```
access-list ACL_OUT extended deny ip any any
```

```
access-list URI extended permit tcp 192.168.10.0 255.255.255.0 any eq www
```

Příkazem *access-list* vytváříme přístupové seznamy ACL (Access List), které obsahují jednotlivé záznamy ACE (Access List Entry).

Každý ACL pojmenujeme, volbou *extended* určujeme rozšířený ACL, volbou *permit* nebo *deny* volíme zda se má provoz na základě dále uvedených argumentů povolit (permit) nebo zamítnout (deny). Dále uvádíme použitý protokol, zdrojovou adresu a masku sítě, cílovou adresu a masku sítě a operátor portu nebo typ ICMP protokolu.

access-list jméno_acl extended permit protokol zdrojová_ip_adresa maska_sítě cílová_ip_adresa maska_sítě operátor_portu log

Volba *host* před adresou označuje jednotlivého hosta, není tedy nutno uvádět masku sítě.

Volba *log* označuje logovaný provoz.

První záznam v *ACL_OUT* povoluje ICMP (Internet Control Message Protocol) na jakékoliv zařízení ze sítě 192.168.10.0 / 24. Dále povolujeme TCP na portech 80, 25, 5353 a UDP na portu 53 z jakéhokoliv zdroje na IP adresu hosta 192.168.10.2.

Access list *URI* nám identifikuje provoz který budeme v další konfiguraci inspektovat na aplikační vrstvě.

```
access-group ACL_OUT in interface outside
```

Příkazem *access-group* svážeme access list *ACL_OUT* s rozhraním *outside* ve směru *in* (provoz směřující dovnitř do rozhraní).

access-group jméno_acl směr interface jméno_rozhraní

```
ip address 192.168.10.7 255.255.255.255
```

```
username Admin pass a0m1n
```

```
aaa authentication ssh console LOCAL
ssh 192.168.10.15 255.255.255.255 outside
```

Příkazem *ip address ip_management_adresa_fw maska_sítě* nastavíme adresu pro vzdálený management firewallu.

Příkazem *username jméno_uživatele pass heslo_uživatele* vytvoříme záznam v lokální databázi firewallu pro daného uživatele.

Příkazem *aaa authentication ssh console LOCAL* definujeme, že k autentizaci uživatele bude použita lokální databáze firewallu.

Příkaz *ssh ip_adresa maska_sítě vstupní_rozhraní* povoluje přístup na firewall pomocí SSH z dané IP adresy a ze strany vstupního rozhraní.

```
class-map CM
match access-list URI
```

Příkazem *class-map jméno_class_mapy* vytvoříme class mapu.

Příkazem *match access-list jméno_access_listu* specifikujeme že tato class mapa je identifikována pokud obsahuje hodnoty specifikované daným ACL.

```
regex URI2 "[oO][nN][lL][iI][nN][eE][hH][rR][yY].*"
regex URI "[xX][cC][hH][aA][tT].*"
```

Příkazem *regex jméno_regulerního_výrazu regulerní_výraz* definujeme regulerní výraz který budeme dále používat při inspekci HTTP protokolu. V tomto případě např. onlinehry.com nebo xchat.cz.

```
class-map type regex match-any CM_REG_URI
match regex URI
```

```
match regex URI2
```

Příkazem *class-map type regex match-any jméno_regex_class_mapy* vytvoříme class mapu typu regulerních výrazů.

Klíčové slovo *match-any* znamená, že kterýkoliv z regulerních výrazů uvedených dále za příkazem *match* identifikuje tuto regex class mapu (logická funkce typu OR).

Příkazem *match regex jméno_regulerního_výrazu* specifikujeme že tato class mapa typu regulérního výrazu je identifikována, pokud obsahuje hodnotu definovanou regulérním výrazem.

```
class-map type inspect http match-all CM_INSPECT
match request header host regex class CM_REG_URI
```

Příkazem *class-map type inspect http match-all jméno_inspekční_class_mapy* vytvoříme class mapu typu inspekce HTTP protokolu. Klíčové slovo *match-all* znamená, že všechny podmínky uvedené za výrazy *match* musí být současně splněny, aby identifikovaly tuto inspekční class mapu (logická funkce typu AND).

Příkazem *match request header host regex class jméno_regex_class_mapy* specifikujeme, že požadavek (request) HTTP protokolu musí mít v hlavičce v poli host hodnotu definovanou class mapou typu regulerního výrazu, aby identifikoval tuto inspekční class mapu.

```
policy-map type inspect http PM_INSPECT
class CM_INSPECT
reset log
```

Příkazem *policy-map type inspect http jméno_inspekční_policy_mapy* vytvoříme inspekční policy mapu protokolu HTTP.

Příkazem *class jméno_inspekční_class_mapy* definujeme class mapu, která identifikuje tuto policy mapu.

reset log je akce, která se má provést, pokud je tato policy mapa identifikována (resetovat spojení a logovat tuto akci)

```
policy-map PM
class CM
inspect http PM_INSPECT
```

Příkazem *policy-map jméno_policy_mapy* vytvoříme policy mapu.

class jméno_class_mapy definujeme class mapu která identifikuje tuto policy mapu.

inspect http jméno_inspekční_policy_mapy definujeme inspekční policy mapu která pokud je identifikována, provede inspekci HTTP protokolu.

```
service-policy PM interface inside
```

Příkazem *service -policy* aktivujeme policy mapu na zvoleném rozhraní.

service-policy jméno_policy_mapy interface jméno_rozhraní

```
policy-map global_policy
```

Globální policy mapa definovaná class mapou *inspection_default* .

```
class inspection_default
```

Class mapa, která je identifikována defaultním nastavením.

```
inspect icmp
```

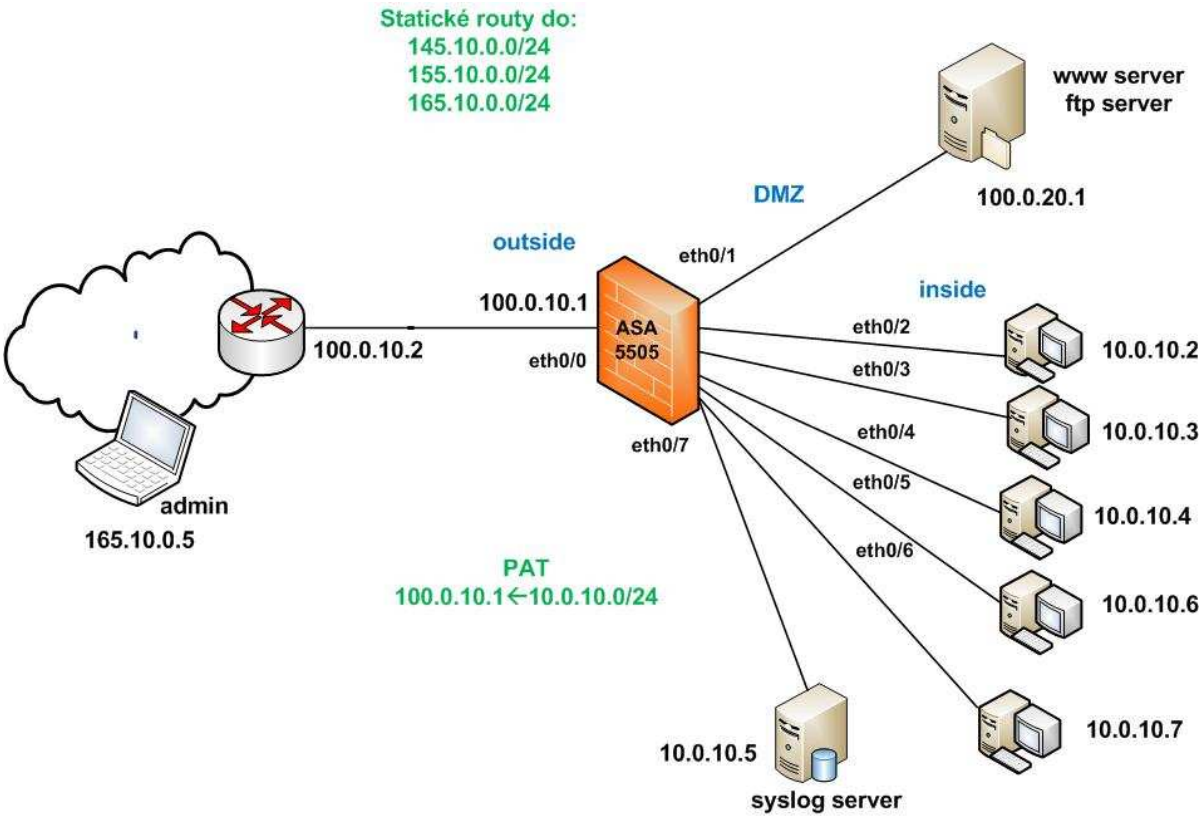
Inspekce protokolu ICMP.

7.2. ASA 5505 v routovacím módu

7.2.1. Popis zapojení ASA 5505 v routovacím módu

V tomto zapojení bude firewall umístněn mezi tři sítě, a to vnější nedůvěryhodnou síť (100.0.10.0/24 - Outside), vnitřní důvěryhodnou síť (10.0.10.0/24 – Inside) a DMZ, ve které bude umístněn WWW a FTP server (100.0.20.0/24). Provoz mezi Inside a DMZ sítí bude omezen tak, že pouze uživatelé Inside sítě mohou posílat pakety do DMZ sítě a ne opačným směrem z DMZ do Inside sítě. Budeme používat překlad adres PAT, kdy všichni uživatelé z Inside sítě budou při komunikaci s Outside sítí maskování za jedinou adresou (adresou rozhrani 100.0.10.1). Omezíme počet TCP spojení z Outside sítě do Inside sítě na hodnotu 1000 a počet nenavázaných spojení na hodnotu 100. Umožníme SSH přístup na IP adresu firewallu pro vzdáleného administrátora. V Inside síti bude umístněn logovací server na který budeme posílat systémové logy firewallu. Nakonfigurujeme routy do vzdálených sítí (145.10.0.0, 155. 10.0.0 a 165.10.0.0) a aktivujeme funkci RPF pro detekci IP spoofingu.

Pomocí časovačů a limitů TCP spojení, omezíme provoz na portu 80, a to z důvodů předcházení útokům typu DoS na WWW server umístněný v DMZ. Povolíme provoz HTTP a FTP z jakéhokoliv zdroje na náš server v DMZ. Také povolíme SSH spojení a všechen provoz od vzdáleného administrátora s IP adresou 165.10.0.5. Uživatelé Inside sítě budou mít přístup mimo Inside síť pouze na portu 22 (SSH).



Obr. 15: Zapojení ASA 5505 v routovacím módu.

7.2.2.Konfigurace ASA 5505 v routovacím módu

```
hostname FW

domain-name mysite.cz

enable password EnA8Le
```

Nastavíme jméno domény a hosta a heslo do enable režimu.

```
interface Vlan100

nameif outside

security-level 0
```

```
ip address 100.0.10.1 255.255.255.0
interface Vlan200
nameif inside
security-level 100
ip address 10.0.10.1 255.255.255.0
interface Vlan300
no forward interface Vlan200
nameif DMZ
security-level 50
ip address 100.0.20.1 255.255.255.0
```

Vytvoříme jednotlivá vlan rozhraní (*vlan100* , *vlan200* , *vlan300*), tato rozhraní pojmenujeme (*inside* , *DMZ* , *outside*), přidělíme jim úrovně zabezpečení (*100 , 50 , 0*) a příkazem *ip address* přidělíme IP adresy a masky sítí.

ip address ip_adresa_rozhraní maska_sítě

Příkazem *no forward interface* zakážeme komunikaci z rozhraní *vlan300* na rozhraní *vlan200* . Vytvoříme částečnou demilitarizovanou zónu, kdy koncová zařízení z *vlan100* nemohou využít zařízení z *vlan300* k vytvoření spojení do *vlan200* .

no forward interface vlan_číslo

```
interface Ethernet0/0
switchport access vlan100
no shutdown
interface Ethernet0/1
switchport access vlan300
```

```
no shutdown
interface Ethernet0/2 - interface Ethernet0/7
switchport access vlan200
no shutdown
```

Jednotlivé fyzické rozhraní přiřadíme do vytvořených vlan (*ethernet 0/0* do *vlan100* , *ethernet 0/1* do *vlan 300* , *ethernet 0/2 – 0/7* do *vlan200*). Rozhraní aktivujeme příkazem *no shutdown* .

```
logging enable
logging trap warnings
logging host inside 10.0.10.5
```

Povolíme logování na firewallu, generované systémové logy posíláme na náš syslog server s IP adresou 10.0.10.5. Úroveň trapů je nastavena na hodnotu *warnings* (4. úroveň).

```
global (outside) 1 interface
nat (inside) 1 10.0.10.0 255.255.255.0 tcp 1000 100
```

Překlad adres PAT (Port Address Translation) při spojení z *inside* rozhraní do *outside* nakonfigurujeme příkazy *global* a *nat* .

Za příkazem *nat* definujeme název rozhraní, ze kterého se bude provoz překládat, následuje *nat_id* , které odpovídá *nat_id* z příkazu *global* a identifikuje adresy, které chceme překládat, když provoz opouští dané rozhraní. Dále uvedeme rozsah IP adres s maskou sítě, které chceme překládat (tzv. reálné IP adresy). Volbou *tcp* můžeme omezit počet současně probíhajících TCP spojení, v našem případě na hodnotu 1000 a počet současně nenavázaných TCP spojení (tzv. embryonic), naše hodnota je 100.

nat (jméno_rozhraní) nat_id reálné_ip_adresy maska_sítě tcp počet_tcp_spojení počet_tcp_embrionic

Za příkazem *global* definujeme název rozhraní na kterém se bude překlad adres provádět, následuje *nat_id* a IP adresa za kterou budou reálné adresy přeloženy (tzv. mapovaná adresa), v našem případě volbou *interface* volíme IP adresu daného rozhraní.

global (jméno_rozhaní) nat_id mapované_ip_adresy

```
access-list ACL_IN extended permit tcp any any eq ssh log
access-list ACL_IN extended deny tcp any any log
```

Přístupovým seznamem *ACL_IN* povolíme SSH spojení.

```
access-list ACL_OUT extended permit tcp any host 100.0.20.2 eq www log
access-list ACL_OUT extended permit tcp any host 100.0.20.2 eq ftp log
access-list ACL_OUT extended permit ip host 165.10.0.5 any log
access-list ACL_OUT extended permit tcp any host 100.0.10.1 eq ssh log
access-list ACL_OUT extended deny ip any any
```

Přístupovým seznamem *ACL_OUT* povolíme provoz HTTP a FTP (File Transfer Protocol) z jakéhokoliv zdroje na náš server s IP adresou 100.0.20.2 v DMZ. Povolíme všechen provoz od vzdáleného administrátora s IP adresou 165.10.0.5 a SSH spojení do sítě 10.0.10.0/24.

```
access-group ACL_OUT in interface outside
access-group ACL_IN in interface inside
```

Svážeme access list *ACL_OUT* s rozhraním *outside* ve směru *in* a *ACL_IN* s rozhraním *inside* ve směru *in* .

```
username AdmiN password a0m1n encrypted privilege 15
aaa authentication ssh console LOCAL
ssh 165.10.0.5 255.255.255.255 outside
```

Vytvoříme uživatele a heslo, autentizace uživatele bude provedena vůči lokální databázi firewallu a povolíme SSH přístup z IP adresy 165.10.0.5 ze směru rozhraní *outside*.

```
route outside 145.10.0.0 255.255.255.0 100.0.10.2 1
route outside 155.10.0.0 255.255.255.0 100.0.10.2 1
route outside 165.10.0.0 255.255.255.0 100.0.10.2 1
```

Příkazem *route* vytvoříme statické routy, za tímto příkazem následuje jméno rozhraní za kterým se síť, ke které routu vytváříme, nachází, následuje IP adresa sítě, maska sítě a IP adresa výchozí brány.

route jméno_rozhraní ip_adresa_sítě maska_sítě ip_adresa_brány

```
ip verify reverse-path interface outside
```

Tímto příkazem aktivujeme funkci RPF (Reverse Path Forwarding), která vytváří obranu proti IP spoofingu, *jméno_rozhraní* je rozhraní, na kterém RPF spustíme.

ip verify reverse-path interface jméno_rozhraní

```
class-map CM_CONN
match port tcp eq www
```

Vytvoříme class mapu, která bude identifikována TCP spojením na portu 80 (HTTP).

```
policy-map PM_CONN
class CM_CONN
set connection conn-max 300 embryonic-conn-max 100
set connection timeout embryonic 0:20:00 tcp 2:00:00
```

Vytvoříme policy mapu a nadefinujeme class mapu která identifikuje tuto policy mapu.

Příkazy *set connection* nastavíme parametry TCP spojení.

Jako maximální počet současných spojení:

set connection conn-max počet_tcp_spojení embryonic-conn-max počet_tcp_embryonic

Nebo časovač po jehož vypršení se spojení nuceně ukončí:

set connection timeout embryonic čas_embryonic_spojení tcp čas_tcp_spojení

Formát času je hh:mm:ss.

```
service-policy PM_CONN interface outside
```

Příkazem *service -policy* aktivujeme policy mapu na zvoleném rozhraní.

```
policy-map global_policy
class inspection_default
inspect icmp
```

Těmito příkazy nastavíme inspekci ICMP protokolu v globální policy mapě, která je identifikována defaultní inspekční class mapou.

8. Testování firewallu

Ačkoliv firewally hrají centrální roli v ochraně sítě a v mnoha případech jsou pouze jedinou obranou proti útokům, systematické testovaní firewallů bylo přehlíženo po mnoho let. Hlavním důvodem byla absence spolehlivé, efektivní a akceptovatelné testovací metodologie.

Existují dva základní přístupy k testování firewallů:

- Penetrační testování
- Testování implementace firewallu

Cílem penetračního testování je odhalit bezpečnostní trhliny dané sítě pomocí běžících útoků proti ní. Penetrační testování obsahuje: sběr informací, průzkum sítě a útoky na cíl. Útoky jsou vykonávány nástroji na odhalení zranitelnosti (jako Nessus, Hping, Nmap, Ettercap atd.), které ověřují firewall a jeho potenciální bezpečnostní trhliny a jejich využitelnost. Penetrační testování je obvykle vykonáváno buď systémovými administrátory, nebo třetí stranou (hackeři, bezpečnostní specialisté), kteří se pokoušejí získat neoprávněný přístup do počítačových systémů.

Testování implementace firewallu se zaměřuje na software firewallu, vyšetřují se chyby (bugs) v implementaci firewallu. Testování implementace firewallu kontroluje, zda software firewallu provádí akce, které má firewall podle nastavených pravidel vykonat, např. pokud pravidlo firewallu má blokovat přijatý paket, ale firewall tento paket přepošle dále, hovoříme o chybě implementace firewallu. Testování implementace firewallu je primárně vykonáváno výrobci pro zvýšení spolehlivosti jejich produktů.

8.1. Penetrační testování firewallu

Pomocí vhodných dostupných nástrojů otestujeme dvě předchozí zapojení firewallu ASA 5505 tak, abychom ověřili funkčnost navržených zapojení. Systémové logy firewallu, které se ukládají na logovacím serveru, nám poslouží k vyhodnocení akcí firewallu. Tyto logy nám také umožňují zpětně zjišťovat zda nedošlo k útokům zvenčí na zařízení v chráněné síti a mapují také chování uživatelů uvnitř chráněné sítě. Tím nám pomáhají při odhalování bezpečnostních trhlin v navržených zapojeních.

8.1.1.Formát zprávy systémového logu

Zprávy generované zařízením ASA začínají znakem % a jsou v následujícím formátu:

% Název zařízení - Úroveň zprávy - Číslo zprávy: Text zprávy

- Název zařízení - ASA nebo PIX.
- Úroveň zprávy - od čísla 0 (emergencies) do čísla 7 (debugging), kdy 0 je nejnižší úroveň generovaných zpráv.
- Číslo zprávy - unikátní šestimístné číslo, které identifikuje systémový log.
- Text zprávy - text popisující události může obsahovat IP adresy, čísla portů nebo uživatelská jména.

8.1.2.Penetrační testování firewallu v transparentním módu

V této části budeme testovat firewall ASA 5505 v transparentním módu. Použijeme nástroje Tracert, Nmap, Ettercap a Nessus.

Zjistíme otevřené porty na firewallu, které jsou dostupné z nedůvěryhodné části sítě při spojení do důvěryhodné části sítě, zmapujeme síťovou architekturu. Pokusíme se provést ARP Spoofing a ověříme, zda správně funguje HTTP filtrování.

8.1.2.1. Traceroute

Nástrojem Traceroute, který slouží k mapování síťové architektury, vypíšeme všechny uzly na cestě k zařízení 100.0.10.2, z důvěryhodné sítě.

Pomocí parametru -I volíme ICMP protokol.

Z výpisu vidíme, že provoz jde přes vychozí bránu 192.168.10.5. Firewall není na tomto výpisu viditelný (je v transparentním módu).

Výpis z nástroje tracert na konzoli hosta 192.168.10.2 :

```
root@student-desktop:/home/student# traceroute -I 100.0.10.2
traceroute to 100.0.10.2 (100.0.10.2), 30 hops max, 40 byte packets
 1  192.168.10.5 (192.168.10.5)  1.694 ms  1.763 ms *
```

```
 2  * * *

 3  * * *

 4  * * *

 5  * * *

 6  * 100.0.10.2 (100.0.10.2)  0.660 ms  0.745 ms
```

8.1.2.2. Nmap

Nástrojem Nmap zmapujeme otevřené porty na zařízení 192.168.10.2, z nedůvěryhodné sítě.

Parametrem -sS volíme techniku skenování TCP Syn, parametrem -PN zacházíme s mapovaným hostem jako s aktivním (neprovádíme kontrolu stavu ICMP dotazem), parametrem -O povolujeme detekci OS na cílovém zařízení, parametr -p definuje rozsah skenovaných portů.

Zdetekovali jsme tři otevřené porty TCP 5353 ,TCP 25 a TCP 80, OS nebyl přesně určen.

Zkrácený výpis z nástroje Nmap na konzoli hosta 100.0.10.2 :

```
root@drahus-laptop:/home/drahus# nmap -sS -PN -O -p20-80 192.168.10.2

Interesting ports on 192.168.10.2:

Not shown: 59 filtered ports

PORT   STATE  SERVICE

25/tcp open   smtp

80/tcp open   http

Running (JUST GUESSING) : Linux 2.6.X (88%), Buffalo embedded (86%), Linksys embedded (86%)

root@drahus-laptop:/home/drahus# nmap -sS -PN -O -p5350-5355 192.168.10.2

Interesting ports on 192.168.10.2:

PORT     STATE    SERVICE

5350/tcp filtered unknown

5351/tcp filtered unknown
```

```
5352/tcp filtered unknown

5353/tcp open     unknown

5354/tcp filtered unknown

5355/tcp filtered unknown
```

8.1.2.3. ARP spoofing

Pomocí nástroje Ettercap a jeho grafické nástavby Ettercap -G jsme se pokusili o ARP spoofing ze zařízení 192.168.10.3. Tento útok nebyl úspěšný.

Systémové logy vygenerované ASA firewallem při ARP spoofingu:

```
%ASA-6-302020: Built outbound ICMP connection for faddr 192.168.10.5/32487 gaddr 192.168.10.3/0 laddr 192.168.10.3/0

%ASA-4-313004: Denied ICMP type=0, from laddr 192.168.10.3 on interface inside to 192.168.10.5: no matching session

%ASA-6-302021: Teardown ICMP connection for faddr 192.168.10.5/0 gaddr 192.168.10.3/32487 laddr 192.168.10.3/32487
```

8.1.2.4. Nessus

Nástrojem Nessus jsme odhalili podle MAC adresy ASA firewallu jeho výrobce.

Výpis z nástroje Nessus:

```
Plugin output:

The following card manufacturers were identified : 00:24:14:5e:5e:bb : Cisco Systems
```

8.1.2.5. Http filtrování

Otestujeme, zda firewall správně filtruje HTTP protokol na základě definovaných regulerních výrazů. Zkusíme přístup na www.onlinehry.cz (194.79.52.193). Firewall toto spojení podle předpokladů resetuje.

Systémové logy vygenerované ASA firewallem:

```
%ASA-7-609001: Built local-host inside:192.168.10.3

%ASA-7-609001: Built local-host outside:194.79.52.193

%ASA-6-302013: Built outbound TCP connection 72 for outside:
194.79.52.193/80 (192.168.10.5/80) to inside:192.168.10.3/39566
(192.168.10.3/39566)

%ASA-5-415008: HTTP - matched Class 22: CM_INSPECT in policy-map
PM_INSPECT, header matched - Resetting connection from
inside:192.168.10.3/39566 to outside0

%ASA-5-304001: 192.168.10.3 Accessed URL 194.79.52.193:/

%ASA-6-302014: Teardown TCP connection 72 for outside:194.79.52.193/80 to
inside:192.168.10.3/39566 duration 0:00:00 bytes 0 Flow closed by
inspection
```

8.1.1.Penetrační testování firewallu v routovacím módu

V této části budeme testovat firewall ASA 5505 v transparentním módu. Použijeme nástroje Tracert, Nmap, Ettercap a Nessus.

8.1.1.1. Tracert

Tracert z hosta 165.10.0.2 na server umístněný v DMZ. Z výpisu vidíme, že poslední odpověď je z uzlu 165.10.0.1(rozhraní routeru). Další ICMP paket neprojde firewallem.

Firewall zabraňuje mapování síťové architektury, zároveň nás ukončení výpisu informuje o tom, že na cestě k serveru je zařízení, filtrující provoz.

Výpis z nástroje tracert na konzoli hosta 165.10.0.2 :

```
root@student-desktop:/home/student# tracert -I 100.0.20.2

traceroute to 100.0.20.2 (100.0.20.2), 30 hops max, 40 byte packets

 1  165.10.0.1 (165.10.0.1)  0.653 ms  0.747 ms *
```

```
 2  * * *

 3  * * *
```

8.1.1.2. Nmap

Na zařízení 165.10.0.2 zmapujeme nástrojem Nmap otevřené porty na zařízení 100.0.20.2.

Parametrem -A zvolíme detekci OS a traceroute.

Našli jsme dva otevřené porty TCP 21 a TCP 80, OS nebyl přesně určen. Máme výpis traceroute s použitím TCP 21.

Zkrácený výpis z nástroje Nmap na konzoli hosta 165.10.0.2:

```
root@drahus-laptop:/home/drahus# nmap -A -sS -PN -p1-1024 100.0.20.2

Starting Nmap 4.62 ( http://nmap.org ) at 2010-04-23 15:08 CEST

Interesting ports on 100.0.20.2:

Not shown: 1022 filtered ports

PORT   STATE  SERVICE VERSION

21/tcp open ftp

80/tcp open http

Warning: OSScan results may be unreliable because we could not find at
least 1 open and 1 closed port

Device type: storage-misc|WAP|general purpose

Running: Buffalo embedded, Isilon OneFS, Linksys embedded, Linux 2.6.X

OS details: Buffalo TeraStation NAS device, Isilon IQ 200 NAS device,
Linksys WAP54G WAP, Linux 2.6.18 (CentOS 5.1, x86)


TRACEROUTE (using port 21/tcp)

HOP RTT  ADDRESS
```

```
1   0.39 165.10.0.1

2   0.96 100.0.20.2

Nmap done: 1 IP address (1 host up) scanned in 7.411 seconds
```

8.1.1.3. IP spoofing

Nástrojem Nemesis zaměníme zdrojovou adresu vyslaného IP paketu.

Parametrem -S (source) nastavíme zdrojovou IP adresu a parametrem -D (destination) zvolíme cílovou IP adresu. Jako zdrojovou IP adresu jsme použili 10.0.10.2 místo skutečné IP adresy 165.10.0.2. Protože firewall za rozhraním outside zná pouze sítě 145.10.0.0, 155.10.0.0 a 165.10.0.0, funkce RPF nepovolí průchodu tohoto paketu firewallem.

Výpis z nástroje Nemesis na konzoli hosta 165.10.0.2:

```
root@student-desktop:/home/student# nemesis tcp -S 10.0.10.2 -D 100.0.20.2

TCP Packet Injected
```

Systémový log vygenerovaný ASA firewallem:

```
%ASA-1-106021: Deny TCP reverse path check from 10.0.10.2 to 100.0.20.2 on
interface outside
```

8.1.1.4. Hping2

Nástroj Hping2 je paketový generátor, kterým provedeme DoS útok na www server v DMZ. Parametrem -c nastavíme počet paketů, které chceme poslat na cílovou IP adresu, parametr -S nastavuje v TCP paketu flag na hodnotu TCP SYN. Parametrem -p definujeme cílový port.

Na firewallu jsme omezili počet současně nenavázaných (embryonic) spojení na portu 80 na hodnotu 100. Každý další TCP SYN nad touto hodnotou je proto firewallem zahozen a zdroji TCP SYN se posíla TCP RST.

Výpis z nástroje Hping2 na konzoli hosta 165.10.0.3:

```
root@student-desktop:/home/student# hping -c 103 -S -p 80 100.0.20.2

HPING 100.0.20.2 (eth0 100.0.20.2): S set, 40 headers + 0 data bytes
```

```
len=46 ip=100.0.20.2 ttl=63 DF id=0 sport=80 flags=SA seq=0 win=5840
rtt=0.7 ms

len=46 ip=100.0.20.2 ttl=63 DF id=0 sport=80 flags=SA seq=1 win=5840
rtt=0.5 ms
```

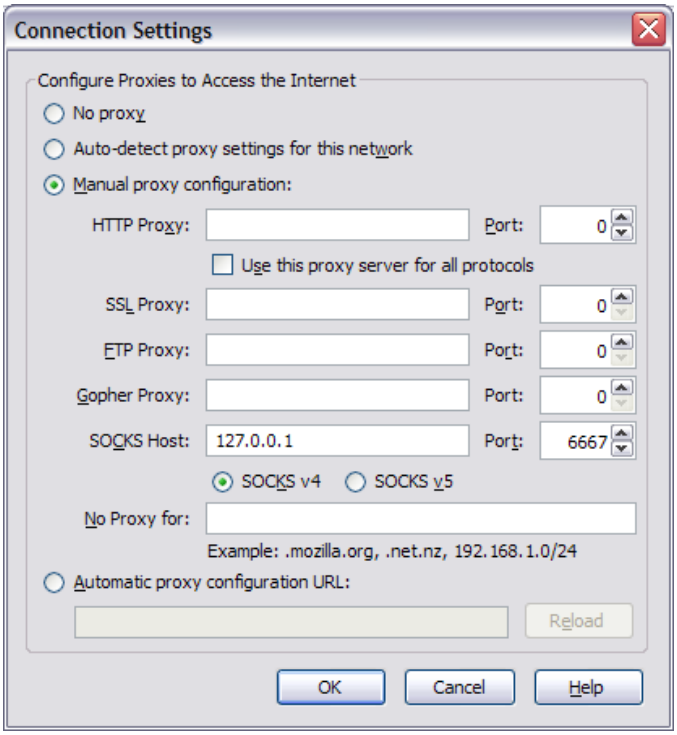
Systémové logy vygenerované ASA firewallem:

```
%ASA-6-201010: Embryonic connection limit exceeded 100/100 for input
packet from 100.0.20.2/80 to 165.10.0.3/2555 on interface DMZ

%ASA-6-106100: access-list ACL_OUT permitted tcp outside/165.10.0.3(2556)
-> DMZ/100.0.20.2(80) hit-cnt 1 first hit [0x13dcafa0, 0x0]

%ASA-6-201010: Embryonic connection limit exceeded 100/100 for input
packet from 100.0.20.2/80 to 165.10.0.3/2556 on interface DMZ

%ASA-6-106015: Deny TCP (no connection) from 165.10.0.3/2554 to
100.0.20.2/80 flags RST  on interface outside

%ASA-6-106015: Deny TCP (no connection) from 165.10.0.3/2555 to
100.0.20.2/80 flags RST  on interface outside

%ASA-6-302014: Teardown TCP connection 15541 for outside:165.10.0.3/2454
to DMZ:100.0.20.2/80 duration 0:00:00 bytes 0 TCP Reset-O
```

8.1.1.5. SSH tunneling

Pomocí SSH tunelování se pokusíme obejít nastavená pravidla firewallu. Uživatelé vnitřní sítě Inside mají povolen provoz do venkovní sítě Outside pouze na portu 22 (SSH). Provoz např. na portu 80 (HTTP) nebo 443 (HTTPS) je zakázán přístupovým seznamem ACL_IN a tím uživatelé ztrácí přístup na www servery ve venkovní síti.

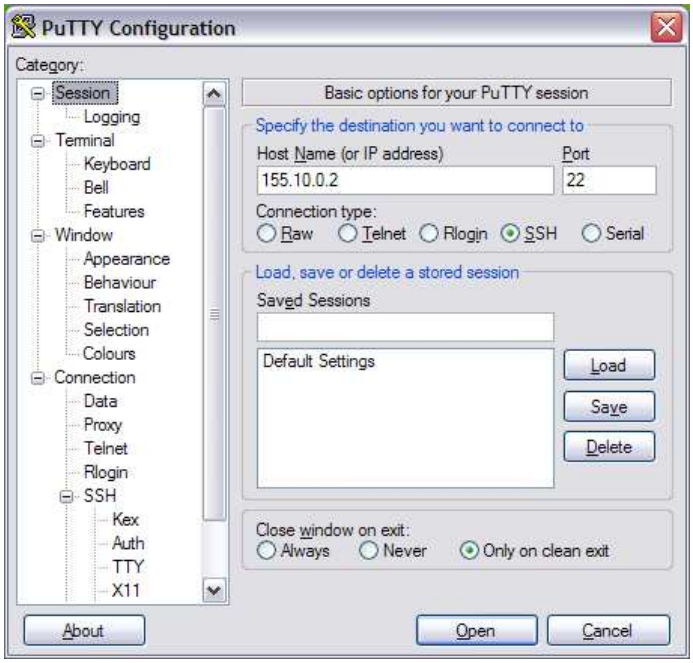
Nastavíme v našem internetovém prohlížeči proxy konfiguraci jako na obrázku Obr. 15.

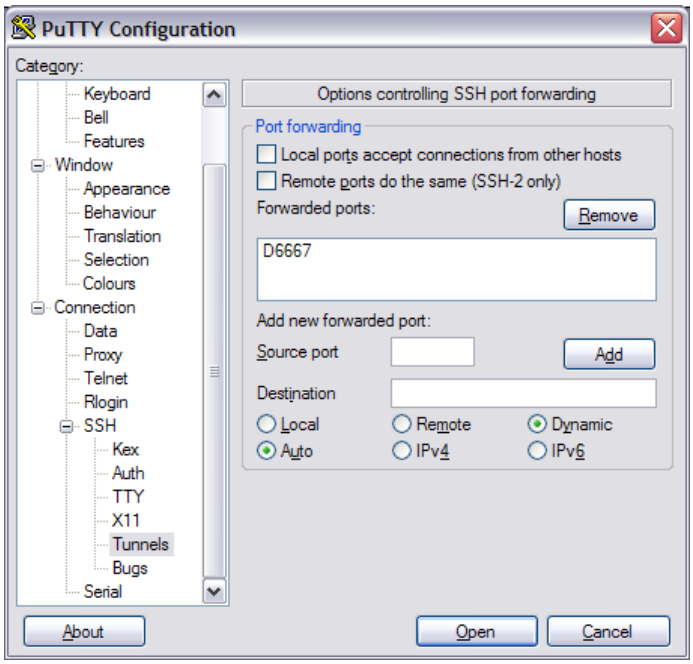


Obr. 16: Příklad nastavení ve Firefox Mozzila

S nástrojem PuTTY vytvoříme SSH spojení k běžícímu SSH serveru (155.10.0.2), který je mimo vnitřní Inside síť a máme k němu přístupová práva (uživatelské jméno a heslo). Do tohoto SSH spojení tunelujeme přesměrovaný port 6667 z našeho internetového prohlížeče. Nastavení PuTTY je na Obr. 16.

Pokud má náš SSH server přístup do internetu na portu 80, můžeme přes toto SSH spojení neomezeně přistupovat na veřejné www servery. Obešli jsme pomocí SSH tunelování nastavená pravidla firewallu.





Obr. 17: Nastavení PuTTY.

9. Závěr

Tato práce se věnovala zabezpečení sítí pomocí firewallu. Popsali jsme si jednotlivé technologie firewallů a jejich historický vývoj. U každé technologie jsme uvedli její nedostatky a výhody, které přináší. Blíže jsme se seznámili se zařízením ASA 5505, u kterého jsme popsali jeho důležité vlastnosti a logiku tohoto zařízení.

Dále jsme navrhli dvě základní zapojení sítě s firewallem ASA 5505. V prvním zapojení byl firewall zapojen v trasparentním módu, v druhém zapojení byl v módu routovacím. V obou zapojeních jsme činnost firewallu zaznamenávali na logovací server umístněný ve vnitřní síti, umožnili jsme vzdálenou správu firewallu přes SSH spojení. Omezili jsme možnost uživatelů vnější sítě vytvářet spojení k uživatelům vnitřní sítě pomocí přístupových seznamů, provedli jsme inspekci ICMP protokolu. V zapojení s ASA firewallem v transparentním módu jsme také inspektovali HTTP protokol, kde jsme zakázali uživatelům vnitřní sítě, vytvářet spojení na námi zakázané webové servery. V zapojení s ASA firewallem v routovacím módu jsme uživatele vnitřní sítě, při komunikaci s uživateli vnější sítě maskovali za IP adresou vnějšího rozhraní firewallu. Nakonfigurovali jsme ochranu proti DoS útokům na webový server umístněný v DMZ a ochranu proti IP spoofingu. Obě zapojení byly plně funkční a firewall plnil funkci aktivního bezpečnostního prvku v síti.

Uvedené zapojení a konfigurace jsme penetračně otestovali. Testování jsme rozdělili na tři části:

- V první části získáváme informace o síťové infrastruktuře a o použitých zařízeních, jak uvnitř sítě, tak o samotném firewallu. Získané informace jsme využili dále k útokům.
- V druhé části jsme provedli několik typů útoků na firewallem chráněná zařízení.
- V třetí části jsme se jako uživatelé vnitřní sítě pokusili obejít nastavená bezpečnostní pravidla firewallu.

Jednotlivé kroky první části testování:

- Mapování architektury sítě

Z výpisu programu Traceroute jsme při zapojení v transparentním módu firewall nezdetekovali. V zapojení s firewallem v routovacím módu nám firewall zabránil mapovat vnitřní architekturu sítě.

- Mapování otevřených portů na firewallu

Nástrojem Nmap jsme u obou zapojení našli otevřené porty na firewallu. V routovacím módu jsme pak na otevřeném portu provedli mapovaní architektury sítě.

- Detekce MAC adresy

Nástrojem Nesus jsme podle MAC adresy firewallu určili jeho výrobce. Tato informace může posloužit k mapování použitých zařízení v síti a zjišťování zranitelností těchto zařízení.

Jednotlivé kroky druhé části testování:

- DoS útok

Provedli jsme pomocí nástroje Hping2 Dos útok na veřejně dostupný webový server v DMZ s použitím TCP SYN paketů. Firewall tomuto útoku zabránil.

- IP spoofing

Nástrojem Nemesis jsme zaměnili zdrojovou adresu paketů a tyto pakety jsme poslali na server v DMZ. Firewall tento provoz zablokoval.

- ARP spoofing

Nástrojem Ettercap jsme se pokusili podvrhnout nepravou MAC adresu výchozí brány uživateli vnitřní sítě v transparentním zapojení. Tento útok firewall zablokoval.

Jednotlivé kroky třetí části testování:

- HTTP filtrování

Vytvořili jsme spojení na webové servery na portu 80. Tento provoz byl firewallem povolen, ale na aplikační vrstvě HTTP protokolu jsme filtrovali jména webových serverů, na které bylo spojení politikou firewallu zakázáno. Při pokusu spojení na tyto servery, firewall správně tento provoz blokoval.

- SSH tunelování

Nástrojem PuTTY a nastavením internetového prohlížeče jsme obešli bezpečnostní politiku firewallu. Zakázaný provoz na portu 80 jsme tunelovali do SSH spojení se vzdáleným SSH serverem mimo vnitřní síť. Z něj jsme pak mohli na zakázaném portu 80 vytvářet spojení.

Podle výsledku těchto testů můžeme ohodnotit správnost našeho zapojení firewallu v síti a jeho konfiguraci.

V první části jsme byli schopni získat určité omezené informace o infrastruktuře, to není ovšem velkým bezpečnostním rizikem.

V druhé části jsme nebyli úspěšní ani v jednom z provedených útoků, firewall je všechny blokoval. Konfigurace a zapojení firewallů byly správné.

V třetí části jsme pomocí SSH tunelování obešli nastavená pravidla firewallu. To je bezpečnostní riziko, které se dá řešit pouze omezením SSH provozu na firewallu. Je nutné povolovat tento provoz pouze jednotlivým uživatelům vnitřní sítě, kteří tuto službu nezbytně potřebují.

Kromě problému s SSH, jsme nezjistili jiné nedostatky v zapojeních a konfiguracích. Po upravení přístupů uživatelů vnitřní sítě k této službě, je možné využívat tyto navržená zapojení k zabezpečení provozu v síti.

Použitá literatura a zdroje

[1] David W. Chapman Jr., Andy Fox. *Zabezpečení sítí pomocí Cisco PIX Firewall.* Brno: Computer Press 2004, ISBN 80-722-6963-1.

[2] Michal Gibbs, Greg Bastien, Earl Carter, Christian Abera Degu. *CCSP SNPA Official Exam Certification Guide Third Edition.* Indianapolis: Cisco Press 2007, ISBN 1-58720-152-6.

[3] Jazib Frahim, Omar Santos. *Cisco ASA: All-in-One Firewall, IPS, and VPN Adaptive Security Appliance.* Indianapolis: Cisco Press 2005, ISBN 1-58705-209-1.

[4] Wes Noonan, Ido Dubrawsky. *Firewall Fundamentals.* Indianapolis: Cisco Press 2006, ISBN 1-58705-221-0.

[5] David Hucaby. *Cisco ASA, PIX, and FWSM Firewall Handbook.* Indianapolis: Cisco Press 2008, ISBN 1-58705-457-0.

[5] David Hucaby. *Cisco ASA, PIX, and FWSM Firewall Handbook.* Indianapolis: Cisco Press 2008, ISBN 1-58705-457-0.

[6] Andrew Whitaker, Daniel P. Newman. *Penetration Testing and Network Defense*. Indianapolis: Cisco Press 2005, ISBN 1-58705-208-3.

[7] *Cisco Security Appliance Command Line Configuration Software Version 7.2.*

URL: http://www.cisco.com/en/US/docs/security/asa/asa72/configuration/guide/asacfg72.pdf

[8] *Top 100 Network Security Tools*. URL: http://www.sectools.org

[9] *Cisco PIX Firewall*. URL: http://www.cs.vsb.cz/grygarek/TPS/projekty/0405Z/PIX/pix.html

[10] *Nessus*. URL: http://www.nessus.org/nessus

Přílohy

konfigurace_ASA_transparent.docx

konfigurace_ASA_routed.docx

logy_pen_testovani_ASA_transparent.docx

logy_pen_testovani_ASA_routed.docx

transparent_nessus_report.html

routed_nessus_report.html